APPLICATION OF SPLINE FUNCTIONS
IN
PICTURE PROCESSING

A Thesis Submitted
In Partial Fulfilment of the Requirements
For the Degree of
MASTER OF TECHNOLOGY

by
S. SRIRAM

to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR

December 1971

POST GRADUATE OFFICE
Submitted on 27.12.71

CERTIFICATE

Certified that this work entitled "Application of Spline Functions in Picture Processing" has been done under my supervision and that it has not been submitted elsewhere for a degree.

Dec.27,1971

K.R. ~~SARMA~~
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur

POST GRADUATE OFFICE
This thesis has been approved for the award of the Degree of Master of Technology (M. Tech.) in accordance with the regulations of the Indian Institute of Technology Kanpur
Dated. 16-2-72

ACKNOWLEDGEMENT

I would like to express my profound gratitude and deep sense of obligation to Dr. K.R. Sarma for his able guidance and constant encouragement throughout the thesis. I am deeply obliged to Dr. B.Prasada for his kind interest and useful suggestions at the most needed time.

I take this opportunity to thank Mr.K.N.Tewari for typing the manuscript with painstaking care. I also wish to acknowledge all others who have helped me in this task.

S. SRIRAM

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A picture can be derived from its "Highs" and "Lows". "Lows" need very little bandwidth for transmission. Hence an efficient coding of "Highs" will ensure a high redundancy reduction in the entire signal domain. This can be achieved by synthesising the "Highs" from the contour information.

A novel method of coding has been suggested for such contour information using Spline Approximation. Application of this encoding procedure to "Synthetic Highs System" results in a large compression in bandwidth required for image transmission. This method of encoding the contour may be directly extended to contour maps, smooth line drawings and similar figures.

Chapter I

INTRODUCTION

1.1 MOTIVATION FOR PICTURE DATA COMPRESSION

There has been a ever-growing demand for transmission of pictorial information in recent years, particularly after the advent of satellites. Introduction of satellites in the scientific world has opened up many new avenues for exploration and revolutionised many areas of application. Consequently, there has been an explosion in the volume of data flow all over the world. Even such diverse fields as mass media communications, space exploration, these days, require enormous amount of information, mostly in the pictorial form, to be transmitted over large distances. Hence, much attention has been given, in recent years to the field of data compression, in particular pictorial data compression.

The reasons for the above is two-fold. First and foremost is the heavy flow of information, to be handled. Besides the problems due to voluminous amount of data, the large distance to be covered imposes stringent restrictions on the quality of transmission by conventional systems. Digital systems like PCM do provide a high quality transmission, unaffected by distance. But, the bandwidth requirements of such a system is unusually

high. Hence, considerable interest has been shown in the last few years towards developing a highly efficient method of coding the image information.

1.2 NATURE OF PREVIOUS STUDIES

The investigations in the field of bandwidth compression could be classified into two categories:

(i) Investigation of highly complicated and sophisticated systems with a view to obtain a very large reduction in transmission rate.

(ii) Study of simple practical techniques giving moderate bit reduction.

The reason for studying highly complicated systems is to obtain a sort of an upper bound on bandwidth reduction ratio that can be achieved without appreciable degradation in subjective picture quality. Even though some of these may not be implemented easily at the present level of 'art', such limitations were not taken into account in view of the objectives specified. Because of the complexity of such systems, these have been studied by means of computer simulation. Further, a computer simulation has the great advantage of flexibility and consequently permits an extensive analysis of the entire system. Most of the studies done so far has been limited to monochrome still pictures since simulation of motion picture requires a large storage capacity. The various extensions of

Synthetic Highs to two dimensions fall under this category.

The Delta modulation scheme and its modified versions like Δ^2 modulation constitute the class of simple,practical systems. Although these have been highly successful for speech transmission, these have not produced satisfactory results in transmitting picture signals.

1.3 BRIEF REVIEW

The thesis under discussion falls under the category of complicated systems mentioned above. Most of the work on this subject, has been done by the Cognitive Information Processing Group at M.I.T.[7,15]. Since the system under study is closely related to Schreiber's Synthetic Highs System[16,17], the following discussion will be confined to papers related to above system only.

1.3.1 Bandwidth Compression of Picture Signals:

Basically, the reduction in bandwidth is possible due to the following factors:

(i) The high bandwidth allowed for conventional TV pictures is mainly needed to reproduce the occasional sharp changes in intensity, occurring near the contours. Since the contours occupy relatively small portion of the picture, there is no need for providing such large bandwidth for transmitting information whose variation is

slow except at relatively small number of points, i.e., statistical properties of the picture should be exploited.

(ii) Human eye does tend to distort the picture in its own way. Hence any attempt to distort the picture without lowering the subjective quality of the picture is welcome, i.e., the visual properties of the eye are also to be taken into account.

Schreiber's philosophy is to treat the picture as a function of several dimensions, spatial as well as temporal. It has been found that such an analysis can result in a much higher reduction than the one obtained by processing the video signal output of the scanner. This is due to the fact that multi-dimensional treatment of the picture is capable of exploiting the interframe and interline correlations.

1.3.2 Synthetic High System:

Experiments indicate that the human eye tends to emphasise the edges in a picture but is relatively insensitive to amount of changes in brightness over edges. On the other hand, in areas where brightness changes slowly, quantization noise is easily discernible. Therefore, the edges and slowly varying part of the picture are to be treated differently. Figure 1.1 shows a block diagram of the system. The video signal, derived from the picture by scanning, is passed through a low pass

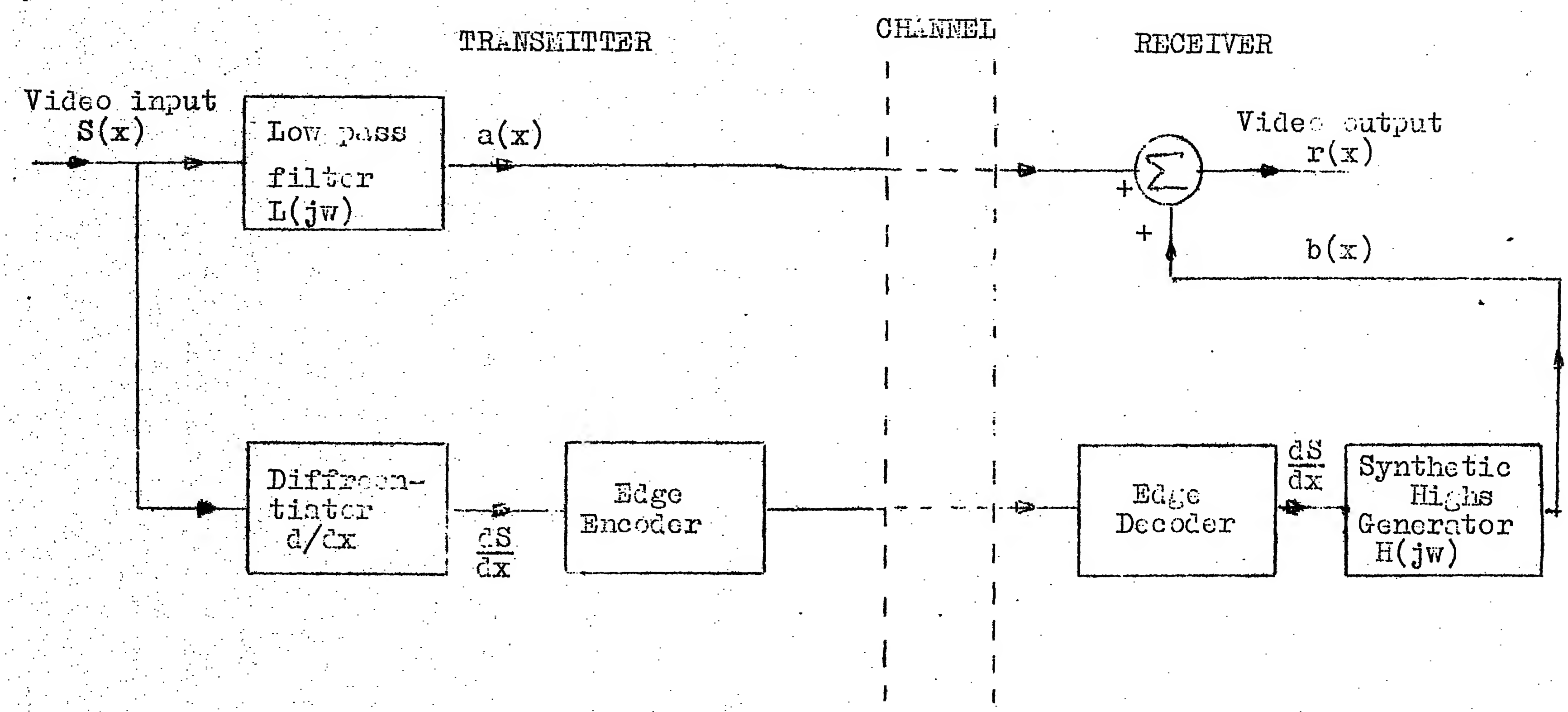


Figure 1.1 SYNTHETIC HIGHS SYSTEM

filter with frequency response $L(jw)$. If the bandwidth of the low pass filter is 1/10th of total bandwidth, the sampling rate has also to be reduced by a factor of 10. The signal $S(x)$ is passed through a differentiator. The output $\frac{dS}{dx}$ contains mainly edge information. If the low pass signal $a(x)$ and the edge $\frac{dS}{dx}$ are transmitted exactly, the original signal $S(x)$ can be synthesised by passing $\frac{dS}{dx}$ through a filter of response

$$H(jw) = (1 - L(jw))/jw \tag{1.1}$$

and adding to it, the low pass signal $a(x)$. This system has been extended to two dimensions by Pan[9] and later on improved further by Graham[6].

In Pan's extension of Schreiber's system, the edges are determined by a local threshold function and information about the position of edge points were transmitted after approximating these by straight lines. No local operator similar to the derivative in single dimension was used. A reduction of 10:1 to 20:1 was obtained. Pictures obtained, however, were of rather poor quality, although still looked reasonable.

Schreiber[17] suggested a mathematical extension of his scheme to two dimensions. The block diagram of such a scheme is given in Figure 1.2. In this system, two different local operators were suggested for edge detection. They are (i) Gradient and (ii) Laplacian.

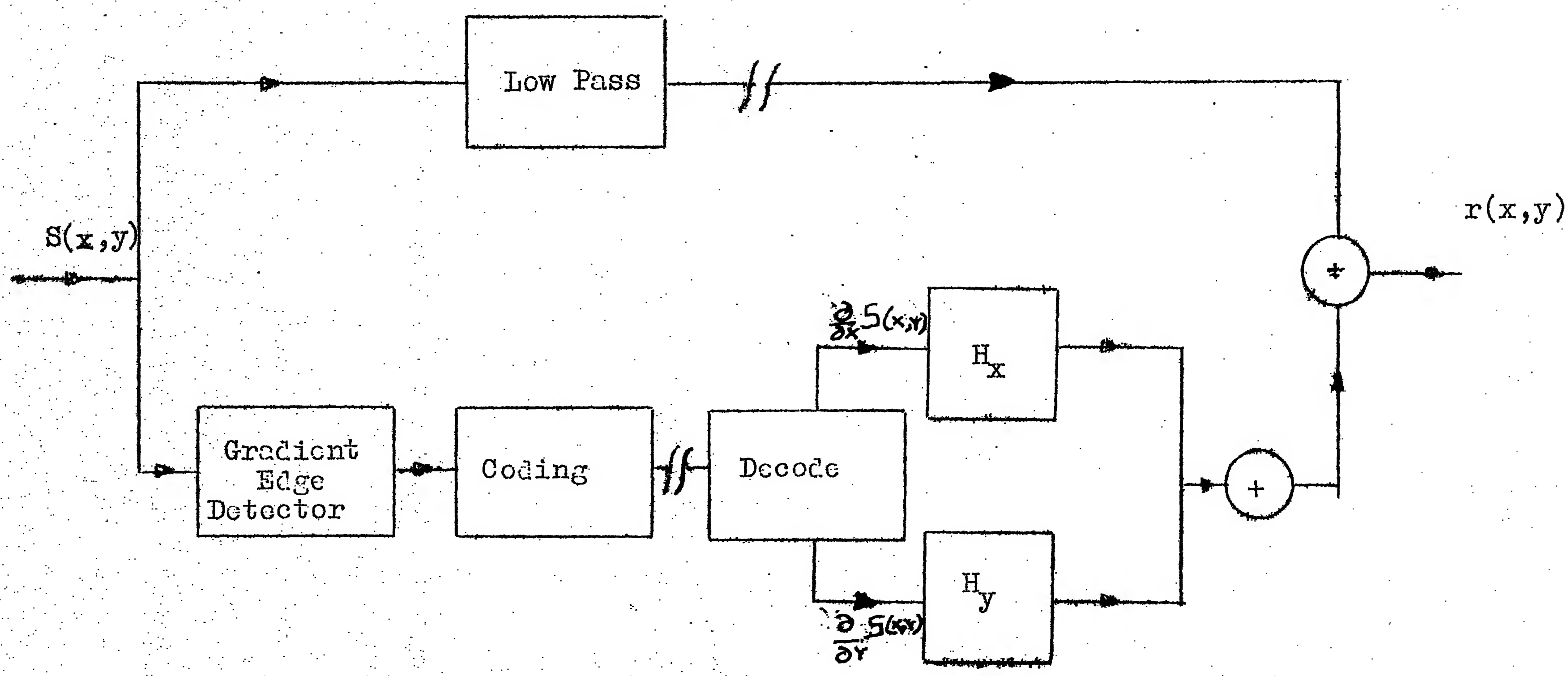


Figure 1.2 TWO DIMENSIONAL SYNTHETIC HIGHS SYSTEM

This system was later simulated by Graham. It was soon apparent that gradient operator is better suited for edge detection, since it is less susceptible to errors due to thresholding than Laplacian. The pictures thus obtained were of extremely good quality with reduction ratios varying from 4:1 to 23:1.

The picture can be obtained directly from the gradient alone. However, in such a system even a small error in reconstruction will generate highly objectionable noise patterns, sometimes masking the picture itself. The presence of LPF reduces such effects (see Appendix I) due to channel noise and other errors.

1.3.3 The Proposed System:

The system under study combines the approximation techniques of Pan and the gradient reconstruction procedure of Graham. However, approximation is done using cubic splines instead of straight lines. This not only gives a substantial reduction in bandwidth but, in addition, smoothens the contours as well. The system has been simulated on IBM 7044 and reduction ratios ranging from 6:1 to 40:1 can be obtained.

1.4 THESIS OUTLINE

The Second Chapter comprises of an introduction to spline functions along with a discussion of various algorithms of spline approximation. In the third Chapter,

the salient features of the proposed systems have been discussed in a detailed manner. The fourth Chapter deals with various aspects of simulation. In the final chapter a critical evaluation of the system performance is presented.

Chapter II

SPLINE APPROXIMATION

2.1 WHAT IS A SPLINE ?

Ever since the days of line drawings, man had to tackle the problem of fairing a smooth curve through a set of points. Often, this was accomplished through the use of long, thin and elastic strips of wood, anchored in place by attaching heavy "ducks" at points along the curve. Smooth lines, so obtained, are the forerunners of what are now widely known as "Spline Functions". Thus, a Spline is just a smooth curve or rather an assembly of pieces so joined as to present the smoothest possible appearance. It is this smooth disjointed nature, more than anything else, which makes it an automatic choice for approximating contours.

2.2 SPLINES AS APPROXIMATING FUNCTIONS

It is often acknowledged that polynomials and similar mathematical functions are inadequate to approximate functions from the physical world. Most of the time, the physical relationships could be more effectively represented by function of a disjointed or disassociated nature than by the more familiar functions of mathematical world. One of the chief advantages of a Spline is that its behaviour in one section can not determine its subsequent

course in another section. Besides , the "Knots" enjoining the various pieces, can be varied in position, thus providing a wide variety of smooth curves. In fact, much of the interest in splines arises from our ability to choose or vary its knots. Above all, spline has an inherent smoothness associated with it as a natural consequence of its minimum curvature property[1].

2.3 MATHEMATICAL REPRESENTATIONS OF SPLINES

A precise mathematical description of spline is to portray it as a piecewise polynomials of degree n that are so connected as to have (n-1) continuous derivatives at the joints, i.e., it belongs to class of functions denoted by $C^{n-1}(0,1)$.

Spline functions are usually defined in terms of their knot set given by

$$0 = k_o < k_1 < \ldots \quad k_m < k_{m+1} = 1$$

The representation of spline function is given as follows:

$$S(\bar{A},\bar{K},x) = \sum_{i=1}^{m} a_i (x-k_i)_+^n + P(x) \tag{2.1}$$

where $P(x)$ is a polynomial of degree n with coefficients a_i, $i = m+1$, $m+n+1$ and $\bar{A}$, $\bar{K}$, $(x-k_i)_+^n$ are given by

$$\bar{A} = (a_1, a_2, \ldots, a_{m+n+1})$$

$$\bar{K} = (k_1, k_2, \ldots, k_m)$$

$$\begin{aligned} (x-k_i)_+^n &= (x-k_i)^n && x \geqslant k_i,\ 1 \leqslant i \leqslant m \\ &= 0 && \text{otherwise} \end{aligned}$$

A spline, with m interior knots, has m+1 polynomials chained to form a single entity. Since a ploynomial requires only (n+1) parameters for unique definition, it is possible to represent the spline uniquely by (n+1)(m+1) parameters. However, it seems from equn.(2.1) that a spline can be represented using only (n+2m+1) parameters. The additional m(n-1) parameters normally needed to specify (m+1) polynomials sections, have been rendered superfluous by the m(n-1) continuity constraints imposed at the joints to ensure the smoothness of the spline. Thus, the continuity constraints not only introduce smoothness in the curve but reduce the number of defining parameters too.

There are many other fforms of representation apart from equn.(2.1). But the most useful one from the point of view of approximation is given below. In this representation the position of the knots, value of the function at these knots plus the (n-1)/2 end conditions at each of the ends are used to specify the spline of an odd degree. That is, $S(\bar{A},\bar{K},x)$ is represented by

$v_{i,o}$ for $i = 0,1,\ldots,m+1$

$v_{o,j}$ and $v_{m+1,j}$ for $j = 1,2,\ldots,(n-1)/2$

where

$$v_{ij} = \frac{d^j}{dx^j} S(\bar{A},\bar{K},x)\Big|_{x=k_i} \tag{2.2}$$

The above representation, by providing direct information about knots greatly simplifies coding procedure.

Also, it requires no more than the minimum possible (n+2m+1) parameters. All these factors render this particular representation highly suitable for coding purposes. However, it has one drawback, viz. the reconstruction of the spline becomes laborious.

To simplify the reconstruction, an intermediate representation which can be obtained from equn.(2.2) is made use of. In this representation, $S(\bar{A},\bar{K},x)$ is represented by the knot set $\bar{K}$ and the matrix

$$\left\{ v_{ij}, \quad \begin{matrix} i = 0,1,\ldots,m+1 \\ j = 1,2,\ldots,(n-1)/2 \end{matrix} \right\}$$

where

$$v_{ij} = \frac{d^j}{dx^j} S(\bar{A},\bar{K},x)\Big|_{x=k_i} \tag{2.3}$$

The representation given in equn.(2.3) gives all the necessary information about each segment of the spline. This permits each segment of the spline to be treated independent of others, thus lessening the computational effort involved in the reconstruction of the spline. It is easy to see that any such computation without arriving at the parameters specified in equn.(2.3) directly from equn.(2.2) is difficult and laborious. The representation given in equn.(2.3) can be easily obtained from equn.(2.2) by solving the appropriate m(n-1) continuity equations. In the case of cubic splines, this

reduces to the problem of solving a system of simultaneous equations, represented by a tri-diagonal matrix.

2.4 SPLINE APPROXIMATION METHODS

There are many algorithms of spline approximation available in the literature[5,12]. These can be broadly classified into two classes. They are:

i) Approximation with fixed knots.

ii) Approximation with free or variable knots.

Although the second one seems to be of relevance to the present problem, the first one will also be considered in detail, since all methods concerning the second case need a fixed knot approximation technique as a subroutine.

2.4.1 Fixed Knot Approximation Problem:

This class of approximation problem may be subdivided into two subclasses based on the norms used for approximation errors. Two of the most popular ones used are the L_∞ norm or the minimax error criterion and the L_2 norm or the least squares error criterion.

Schumaker[18] has given a computational method for approximation with L_∞ norm based on exchange process of Remez variety. Such an approach which assumes an error curve of "normal" form with n+m+1 alternating extrema suffers a disadvantage because of the non-Chebyshevian nature of the approximating function. For some functions,

the best approximations have fewer than n+m+1 alternative extrema. Perhaps a greater drawback is observed in the form of additional error oscillations when joints are near their optimal locations.

Barrodale and Young[2] have used a linear programming approach, which though closely related to exchange processes of Remez variety, is not based explicitly on any characterization process and avoids these difficulties.

J.R. Rice and Carl De Boor[3] have evolved a procedure for L_2 approximation which is much more elegant than any of the methods described above. He has treated the problem as a linear approximation problem and solved it using an orthogonalization procedure.

Powell[11] and Lawson[8] have also dealt with the above problem and a concise discussion of various methods has been presented by Esch and Eastman in their paper[5].

Out of the methods described, Rice's procedure is extremely economical in time. Because of the orthogonalization procedure it can be so manipulated that variation of a particular knot will affect only one orthogonal function and hence, the new error could be obtained from the error at the previous stage of approximation. In all the other methods, the whole procedure has to be completely repeated even for a small variation of a particular knot.

This in-built facility is particularly useful in saving computer time when used as a basis for free knot approximation. Because of this advantage, the algorithm suggested by Rice has been preferred to others for use in the proposed system.

2.4.2 Fixed Knot Approximation by Orthogonalization:

The problem is to determine $\overline{A}^*$ such that

$$\int_0^1 (f(x) - S(\overline{A}^*, \overline{K}, x))^2 \, dx \qquad (2.4)$$

is minimised over the space of vectors $\overline{A}$.

This can be treated as a linear approximation problem and there exists a solution for such problems of approximation using orthogonalization procedure. In this method, the problem is reduced to one of finding a set of orthonormal functions which will form a basis for the given class of approximating functions. It has been shown by Rice[12] that such a basis exists for spline functions. If we denote the set of orthonormal splines by

$$O_j(\overline{K},x), \quad j = 0,1,\dots,n+m$$

For $j \leq n$ $\quad O_j(\overline{K},x) = P_j(x)$

where $P_j(x)$ is the Legendre polynomial of degree j in $(0, 1)$.

For $j > n$ $\quad O_j(\overline{K},x)$, $j = n+1, \dots, n+m$

are given by the equations

$$\int_0^1 O_j(\overline{K},x)\ O_k(\overline{K},x)\ dx = \delta_{jk} \tag{2.5}$$

Thus starting from Legendre polynomials the whole set of orthonormal splines can be determined, using Gram-Schmidt procedure. The optimal approximating spline is given by

$$S(\overline{A}^*, \overline{K}, x) = \sum_{j=0}^{n+m} a_j^*\ O_j(\overline{K}, x) \tag{2.6}$$

where $a_j^* = \int_0^1 f(x)\ O_j(\overline{K}, x)\ dx$

2.4.3 Variable Knot Approximation Problem:

This is basically a nonlinear approximation problem. Here the problem is to determine the position of the knots so as to minimise the error criterion. Once a fixed knot routine, giving the requisite approximation along with the error for a given set of knots, is available, the problem can be reduced to minimization of a nonlinear function which maps any knot set to the corresponding approximation error.

Schumaker[18], Esch[5] and Rice[4] have all tackled this problem in their own ways. But the discussion here is restricted to Rice's algorithm due to reasons already mentioned in 2.4.2.

The problem is to determine an optimal $\overline{K}^*$, $\overline{A}^*$ such that for any given f(x) on (a, b) and $\epsilon > 0$

$$\int_a^b (f(x) - S(\bar{A}^*, \bar{K}^*, x))^2 \, dx \tag{2.7}$$

is minimized with minimum possible number of knots. In other words, $\bar{K}^*$ as well as $\bar{A}^*$ should be determined completely, their dimensions as well as their values.

It has been found theoretically that a solution to the above problem is feasible. In support of this, we have the following theorem (Ref.12, pp.143):-

Set $$E_{n,m}(f) = \| f(x)-S(\bar{A}^*,\bar{K}^*,x) \|_2 \tag{2.8}$$

where the spline $S(\bar{A}^*,\bar{K}^*,x)$ is a best L_2 approximation to $f(x)$ with degree n and m variable knots, then either

$$E_{n,m}(f) = 0$$

or $$E_{n,m+1}(f) \leq E_{n,m}(f) \tag{2.9}$$

Thus, in theory at least, it is possible to achieve the approximation to any desired degree of accuracy with sufficient numbers of knots. However, such an approximation with minimum number of knots is of particular relevance here. This is what Rice and De Boor have attempted to achieve by their computational algorithm.

2.4.4 Rice's Method :

J.R.Rice and Carl De Borr[4] have suggested a procedure for cubic spline approximation with minimum number of free knots. This procedure requires as a basis a set of

orthonormal functions to be determined. First n+1 (here, 4) normalized Legendre polynomials are obtained and f(x) is replaced by

$$f(x) - \sum_{j=0}^{3} a_j^* \, O_j(x) \qquad (2.10)$$

The remaining O_{3+j} $(\overline{K},x)$, $j = 1,2,\ldots,m$ are determined so that $O_{3+j}(\overline{K},x)$ involves only the j leftmost knots. This makes it possible to economize on computations later. The error can be determined as

$$e(x) = f(x) - \sum_{j=0}^{n+m} a_j^* \, O_j(\overline{K},x) \qquad (2.11)$$

Given an initial set of knots, they are improved cyclically in order to minimise the error. The cycle starts with rightmost knot and working to the left, each is varied so as to reduce the error as a function of this knot. This cyclic procedure is continued till suitable convergence criterion is met. The criterion of a particular knot is made using discrete Newton method. While varying a single knot k_i, a temporary orthogonal basis for splines, not involving k_i, is obtained. This basis is used along with one rothogonal basis function $O_i(x)$ involving k_i. Hence, only this last function changes during the discrete Newton method and one can replace the e(x) by

$$e(x,k_i) = - \int_a^b f(x) \, O_i(x) \, dx \qquad (2.12)$$

However, in this procedure care should be taken against crowding of knots, aliasing of knots etc.

Under normal circumstances if the error criterion is not met with the initial number of knots, additional knots are introduced. They are included one at a time and optimized cyclically till the required error criterion or some other exit criterion is met. A point is determined, at the end of each stage, where error is maximum and the new knot is introduced midway between the two knots bracketting the point of maximum error. The algorithm may have to be terminated sometimes even before sufficiently small error is obtained, owing to crowding of knots around a particular point. A similar exit is forced when an unusually large number of knots are required to approximate the given curve. Accuracy of this method, like most of the other optimisation methods, is very much sensitive to the initial choice of knots. This is the major disadvantage of this method. However, this disadvantage is common to all the other methods as well.

Although this algorithm can be extended to splines of higher degree, it is not advantageous to use higher degree splines because of the larger number of parameters required to represent them. Also, the complexity of the procedure increases enormously even for simple problem with fixed knots. Hence only cubic splines have been used in the system under study.

Chapter III

PROPOSED SYSTEM

3.1 INTRODUCTION

This system combines the salient features of Pan's and Graham's schemes. Pan[9] demonstrated that contours can be coded efficiently by proper segmentation, followed by straight line approximations of the various segments. However, Pan adopted an "ad hoc" procedure for reconstructing the picture. Consequently, he succeeded in only producing a picture which could not quite meet the requirements, demanded of it.

Graham[6] subsequently developed an elegant but simple method of reconstructing the picture from its contour information. This scheme could not only impart the requisite quality to the received image but achieve this with ample reduction in bandwidth, as well. Although Graham exploited the connectivity of the picture contours, he did not make use of the natural "smoothness" inherent to the contours of the picture. Nor did he effectively use the correlation between the direction of contours and the gradient direction in subsequent coding of the contours, prior to the actual transmission.

In the proposed system, the Graham's reconstruction procedure has been incorporated with a view to obtain good quality pictures. At the same time an effort has been made

towards exploiting the "smoothness" of the contours which renders it suitable for coding by approximation. Also, a scheme is developed for extracting the part of the gradient information from the approximated contours themselves.

3.2 SALIENT FEATURES

Since the system differs from Graham's scheme mainly in coding of the contour information, only that portion will be considered here.

A simple block diagram (Figure 3.1) illustrates the various parts of the system. The first block forms the contour tracer which extracts all the gradient contours in the picture and sends the position information of the contour and the gradient along these curves separately. The position information is then subjected to a pre-smoothing process which renders it suitable for spline approximation. The parameters of approximation are then fed along with the coarsely quantised gradient information to the encoder for proper coding prior to the actual transmission.

The receiver shown in Fig.3.2 reconstructs the splines approximating the contours and organizes the gradient along these reconstructed curves. The gradient map of the picture, so obtained, is passed through an appropriate high pass filter to be added on, at a later stage, to the defocussed image received at a much less sampling rate. A detailed discussion of each subsystem is given below.

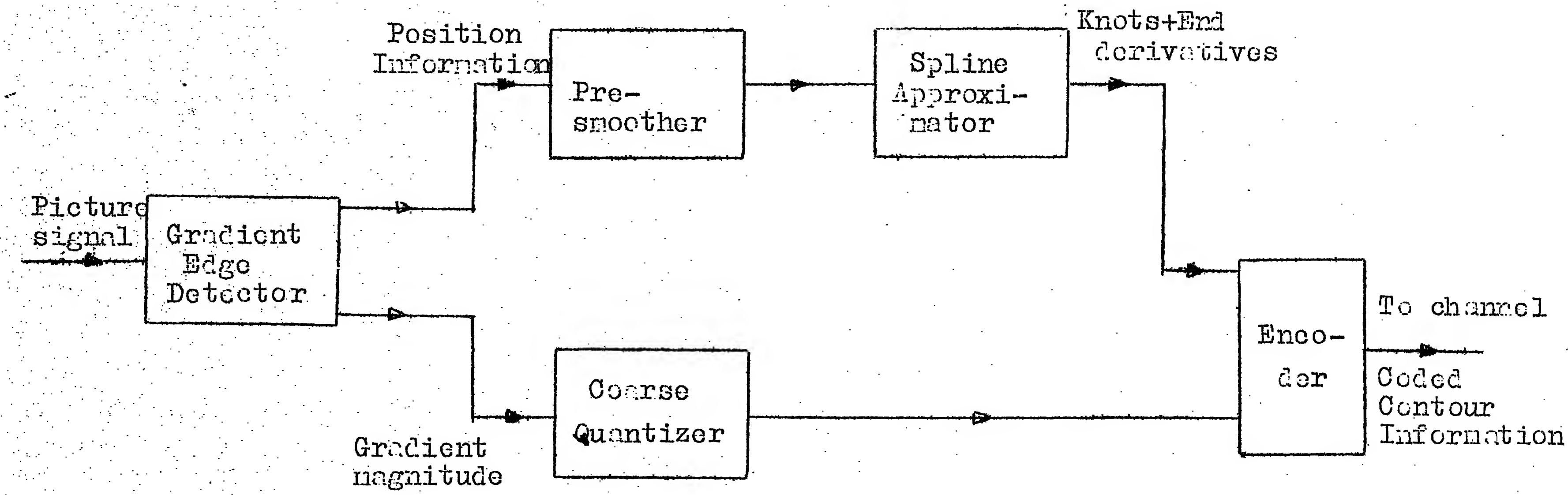


Figure 3.1 CONTOUR - CODER

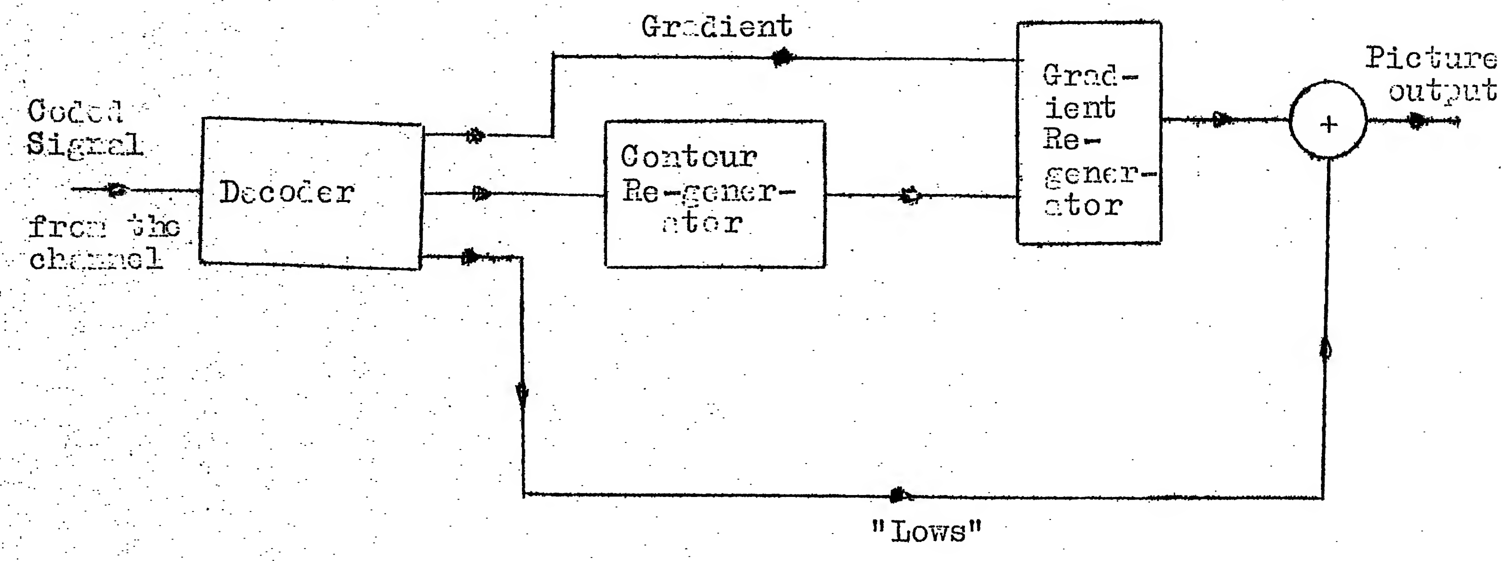


Figure 3.2 RECEIVER

3.2.1 Contour Extraction:

The contours are extracted using the same procedure adopted by Graham[6]. Gradient of the picture signal is calculated for every picture element. A threshold is set for the starting point of the contour. Once this is realized, a search is made among its 8 neighbours (Figure 3.3) using another threshold. This process is continued till there is no continuity in the contours. This way, certain spurious contours due to noise can be eliminated by leaving out very short ones. However, this is not the only algorithm available. Rosenfeld et.al.[13] have suggested a highly effective edge detection technique. Further in the case of digital pictures the gradient can be defined in innumerable ways. Pingle[10] and Sakal[14] have used complicated versions of such definitions to detect edges in noisy pictures. But the one Graham had used, viz. arithmetic difference of intensity values at adjacent points, is extremely simple and easy to implement. Since the study has been done on pictures with very low noise, it was not necessary to go in for highly complex algorithms.

This method of contour extraction needs slight modification before its implementation. Because of the digitization of the picture the contours of the picture, instead of being perfectly smooth, consists of a large number of small edges, both vertical and horizontal. Therefore, it is logical to represent this contour

1 X	2 X	3 X
8 X	0	4 X
7 X	6 X	5 X

0 - EDGE POINT

X - NEIGHBOUR

Figure 3.3 AN EDGE POINT AND ITS NEIGHBOURS.

X - EDGE POINT OF CONTOUR 1

0 - EDGE POINT OF CONTOUR 2

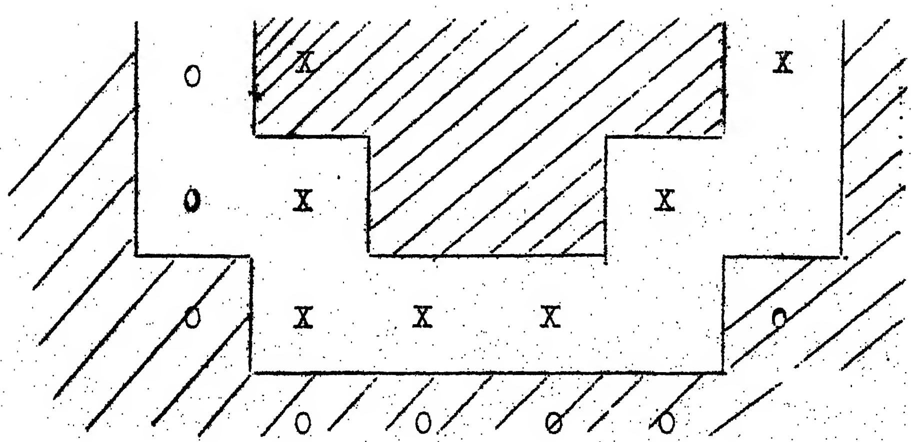


Figure 3.4 CONTOUR REPRESENTATION USING PICTURE ELEMENTS.

in terms of these small edges. However, Graham for the purpose of achieving high coding efficiency has chosen the picture element as a basic unit constituting the contours. This method of representation, in some situations, causes a sort of spread in the contours, leading to a mix-up of edge points in adjacent contours (Figure 3.4). Moreover, since the extracted contour has to be approximated, the particular representation used, should only provide a thin contour instead of high coding efficiency. Hence the small edges constituting the contours have been chosen as the basic unit. These edges are represented in terms of one of their ends as shown in Figure 3.5. The particular end used in the representation of the contour will hereafter be referred to as "node".

3.2.2 Approximation of the Contours:

It may be noted that the contours extracted from the picture have to pass through a device which approximates the extracted contours by splines to the required degree of accuracy and gives as output, the various parameters of approximation. The advantages of using splines lies in its smoothness and its ability to represent the contours in an efficient manner. But the spline functions, for that matter any mathematical function, can approximate only single valued functions. On the other hand, closed contours are very likely to occur quite often in a picture. This necessitates

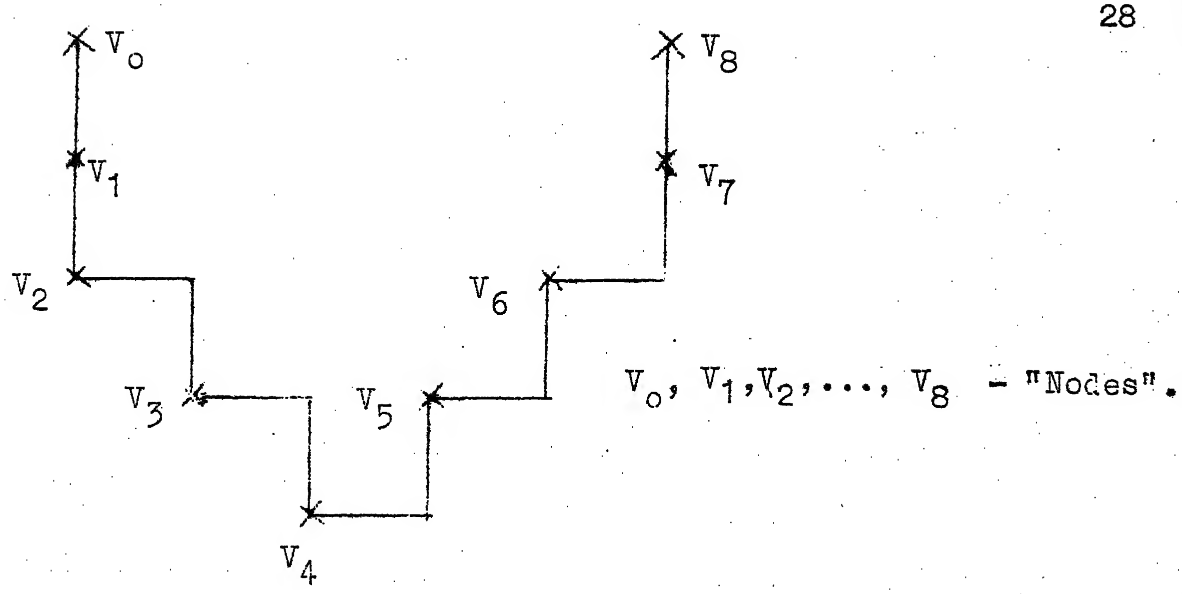


Figure 3.5 CONTOUR REPRESENTATION IN TERMS OF NODES.

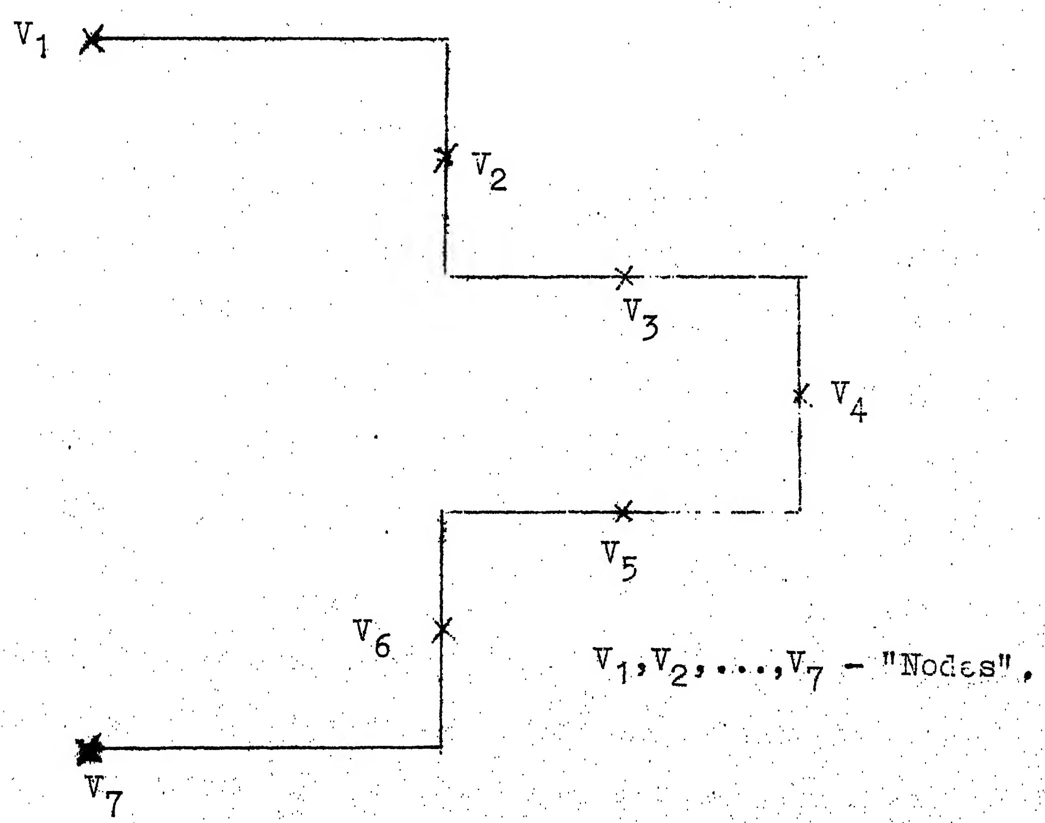


Figure 3.6 REPRESENTATION OF FLAT PORTIONS BY NODES.

some amount of pre-processing of the contours derived from the picture. The pre-processing should be such as to retain the position information as well as provide this in the form of single valued functions. This can be achieved in one of the following ways:

(i) Approximation after segmentation: In this method, the contour is divided into many segments in such a way that in each segment the x-co-ordinate of the nodes constituting the contours are either non-decreasing or non-increasing. Then each of these segments is later smoothed so that no two consecutive nodes representing the contour have their x-co-ordinates equal. This is achieved by choosing the proper ends v_1 and v_2 of horizontao edge and the succeeding vertical edge as nodes (Figure 3.5) or choosing the node lying in the middle of horizontal portion to represent the flat portions of the segments as shown in Figure 3.6. In case the flat portions occur in either the beginning or the end of the segment, the appropriate end of such flat portions are chosen as shown in Figure 3.6.

(ii) Parametric approximation: In this method, each co-ordinate is expressed as a function of contour lengths, a monotonically increasing parameter and approximation procedure is applied to the newly obtained functions. This method obviously suffers from the disadvantage that each curve has to be effectively approximated twice besides

needing double the number of parameters for representing the approximation. Also, in the case of digital pictures where contours are represented by vertical and horizontal edges, there is no proper definition of length of the contour. Perhaps, this difficulty can be overcome by defining the length of any segment to be proportional to the number of small edges constituting it. But this does not solve the problem either, since it has been observed that the reproduction of such approximated contours is not quite satisfactory. This is probably due to the digitised nature of the measure chosen for length. Actually in the process of the parametric approximation of a digital contours, the smoothness of the contour is not retained , thus loosing the very basic property required of spline approximated functions. Moreover, the spline approximations are highly effective for contours of medium length. In order to achieve this efficiency, long contours may have to be any way segmented. Thus for very long contours, we are not really avoiding the segmentation procedure even by this method.

In view of the above mentioned reasons, the segmentation approach has been adopted for the system under study.

3.2.3 Gradient Quantization:

The gradient information of the picture is very vital for a satisfactory image reconstruction. Fortunately,

the human eye is not sensitive to the slow intensity changes along the contour. Moreover, even at contour points, it tends to emphasize only the change in brightness. In fact, the eye is relatively insensitive to the magnitude of these changes. Consequently one can do with a relatively coarse quantization of the gradient magnitude. However, while reconstructing the picture, its quality may be improved, if the gradient is smoothed along the contour. For this purpose splines will be highly useful. Further because of the nature of coding of the contours, the gradient direction can be derived directly from the orientation of the corresponding edge since the gradient direction should always be perpendicular to the edge orientation.

3.2.4 Receiver:

The receiver part of the system is illustrated in Figure 3.2. The functions of its various subsystems have already been briefly discussed in Sec. 3.2.

It has been found, based on the results of the simulation that the picture reconstructed from coded signal is fairly good to look at and has no particularly discernible defects.

Chapter IV

SIMULATION STUDIES

4.1 SOME ASPECTS OF SIMULATION

The entire system has been studied through simulation on IBM 7044. The various aspects relating to the choice of the different parameters of the system will be discussed in this portion of the chapter.

4.1.1 Low Pass Filter:

This forms an integral part of the transmitter. This portion has been simulated using a routine called 'LOWFIL'. Given the input image samples as well as the filter response samples, 'LOWFIL' merely calculates the convolution between the two, i.e., if $h(x,y)$ is the filter response and $b(x,y)$ is the picture signal, then the output of 'LOWFIL' is given by

$$l(x,y) = \iint b(x-p,y-q)\, h(p,q)\,dp\,dq \qquad (4.1)$$

In digitized form, this may be expressed as

$$l_{jk} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} b_{j-a,k-b}\, h_{a,b} \qquad (4.2)$$

$$j,k = 0,1,\ldots,n+m-1$$

where $m \times m$ is the size of the filter response matrix and $n \times n$, that of the picture signal matrix.

However, one does not need all the $(n+m-1)^2$ samples. It is enough to transmit n^2 samples, corresponding to the picture image points, since these n^2 samples are enough to reproduce the image signal. Other things being equal, the choice of the linear filter is mainly dictated by considerations of computer time. It is well known that it is advantageous to use filters with separable kernels, particularly, when the conventional convolution methods are employed. Further, the above algorithm proves to be even faster than ones using FFT techniques, when the filter samples are small in number. With this view in mind, a Gaussian function is chosen as the low pass filter, i.e.

$$l(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(- \frac{x^2 + y^2}{2\sigma^2}\right) \tag{4.3}$$

Although the filter function extends to infinity on either side of the origin, the sample values of this filter fall very rapidly to zero, after first few samples. Hence the filter can be truncated after a finite number of samples without appreciable loss in accuracy. In our case, the cut-off point is chosen to be 4σ and the significant samples are normalized so that

$$\iint l(x,y)\, dx\, dy = 1 \tag{4.4}$$

where $l(x,y)$ is the low pass filter response.

Taking the Fourier transform

$$l(w_r) = \exp\left(- \frac{\sigma^2 w_r^2}{2}\right) \qquad (4.5)$$

where w_r represents the radial spatial frequency.

Since more than 90% of the energy is at spatial frequencies less than w_{rmax}, where w_{rmax} is given by

$$w_{rmax} = \frac{1.516}{\sigma} \qquad (4.6)$$

this itself, can be chosen as the cut-off frequency. Since the sampling theorem holds in two dimensions, a sampling rate of $2f_{max} = w_{rmax}/\pi$ will be sufficient for transmitting the "Lows".

For $\sigma = 4$, the sampling rate $= \frac{1.516}{4\pi}$

That is, if a 64^2 picture is to be transmitted, its 'Lows' need only 8^2 samples. Further, because of the two dimensional interpolation used, it should be possible to quantize it with fewer than 6bits/sample if the original picture is quantized at the same rate.

The choice of in effect determines the number of filter samples as well as the cut-off frequency of the filter. A low cut-off frequency would mean a substantial gain in the number of bits required for transmitting the lows. But, this effectively means a higher value of and hence a larger number of samples are required to represent the filter response, thereby reducing the

speed of the routine. This, however, should not be a matter of concern, if we confine ourselves to simulation of the corresponding analog system. On the other hand, if one chooses to study the behaviour of the digital version of the proposed system with a view to better the existing PCM image transmission, this factor has to be taken care of. Also, the reduction, so obtained, should not be at the cost of any deterioration in picture quality. Graham has stated in his paper, that the value of $\sigma = 4$ gives satisfactory results.

4.1.2 Contour Extraction:

The algorithm for contour tracing has already been explained in the previous chapter. The routine "CONTOR" achieves the various functions required of the contour tracer. One very important parameter to be chosen is the threshold set up for contour selection. Basically, the entire scheme is based on the philosophy that the high frequency portions of the signal occurs only at limited places of the pictures, namely contours. However, due to quantization and other sources of noise, a noisy point may be mistaken for a contour point, if the threshold is not sufficiently high. On the other hand, if it is made very high, one is likely to miss some details altogether. Further, a part of the information has already been sent by 'Lows'. Therefore, the cut-off frequency of the filter which determines the information already

transmitted should be closely related to the threshold for contour selection, which selects the additional information necessary for reproduction. A rough analysis done in Appendix II supports this view.

Since the processing has been performed only on two level pictures, the choice of the threshold did not pose any problem. It is worthwhile to note that Graham was able to reconstruct reasonably good quality pictures using a contour start threshold of 25 and edge point threshold of 6 for 256 level pictures.

4.1.3 Contour Approximation:

The algorithm used for the spline approximation has been dealt with in detail in the Chapter II. The routine 'APPROX' performs this function. However, one important aspect of it, which was not covered in Chapter II, will be discussed here. As already explained, for a spline reconstruction, apart from the information about the knots, two more parameters usually in the form of end derivatives have to be specified. Since the derivatives can assume any value from $+\infty$ to $-\infty$, the problem quantizing it is not straightforward. One way of achieving this is to quantise the angle of inclination, which ranges from $+90°$ to $-90°$. However, that does not fully solve the problem. If a linear quantization is used in such a situation, there is a large error in the derivative

when the inclination is close to right angles. This was observed while studying the simulated system on the computer and a remedial scheme was tried. According to this scheme, the range close to 90° is finally quantized and the rest of the range is quantized in a coarser fashion so that the error is reduced even in extreme cases with more or less the same number of levels as before.

4.1.4 High Pass Filter:

This is used for reconstructing the high frequency portion of the signal from the gradient map of the picture. This forms a part of the receiver. Its design is solely determined by the low pass filter used at the transmitter (see Appendix III). This part of the system is taken care of by the routine 'HIFIL' which directly computes the requisite response using the conventional convolution method. The speed of this routine can be improved by adopting the FFT approach.

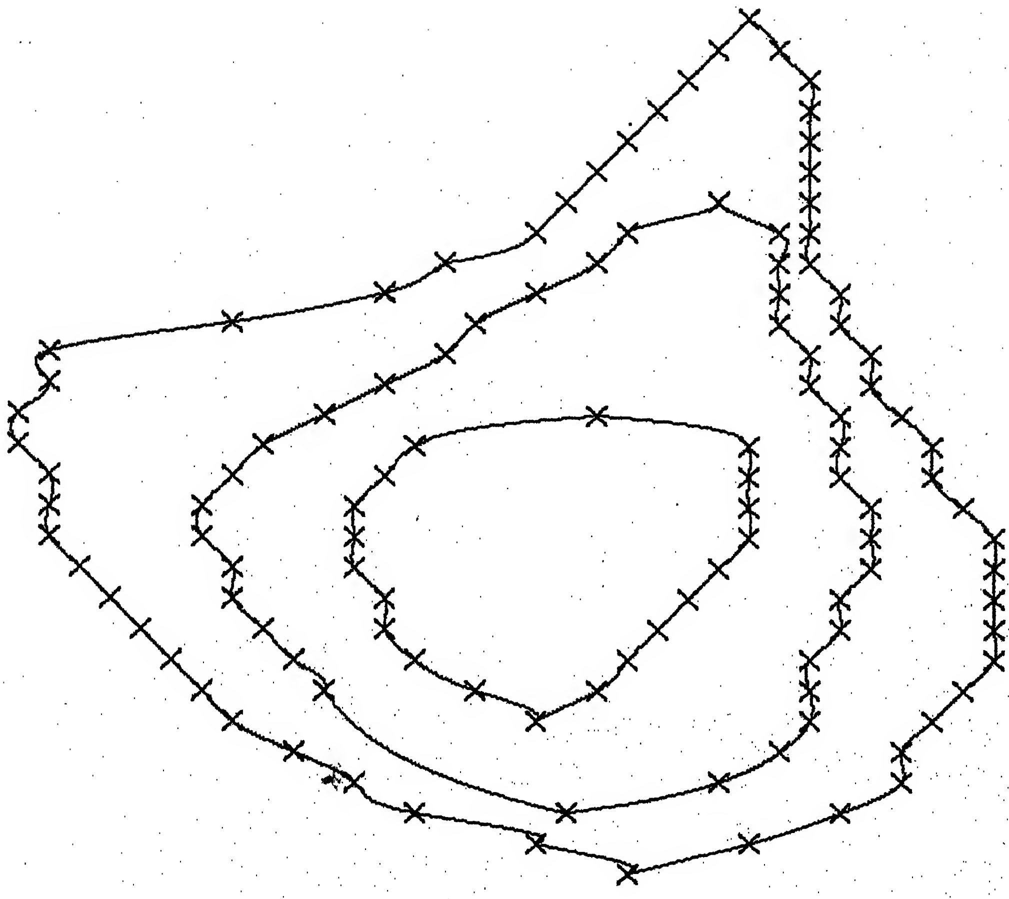
4.2 STUDIES ON CONTOUR APPROXIMATION

The effectiveness of the approximating routine has been tested under various conditions. Two different sets of data have been used in all these experiments. Two weather maps, each consisting of three contours, in the digitised form constitute the data. The first study was done to compare the results using the two

approaches of approximating the contour. The results of this study have been given in Tables 1 and 2 and Figures 4.1 to 4.6. The results clearly indicate that the segmented approach is far superior to the parametric method both in respect of the number of parameters of approximation and the closeness of the approximation.

A study of quantization of end conditions was also done on these two maps and the results shown in Table 3 and Figures. 4.7 and 4.8. The results generally indicate that a 6 bit quantization gives satisfactory results under normal conditions. However, for curves requiring steep end conditions, some error is inevitable. This tendency can be reduced to a great extent by using a high resolution data of the same map. The results obtained are listed in Table 3 and shown in pictorial form in Figure 4.9.

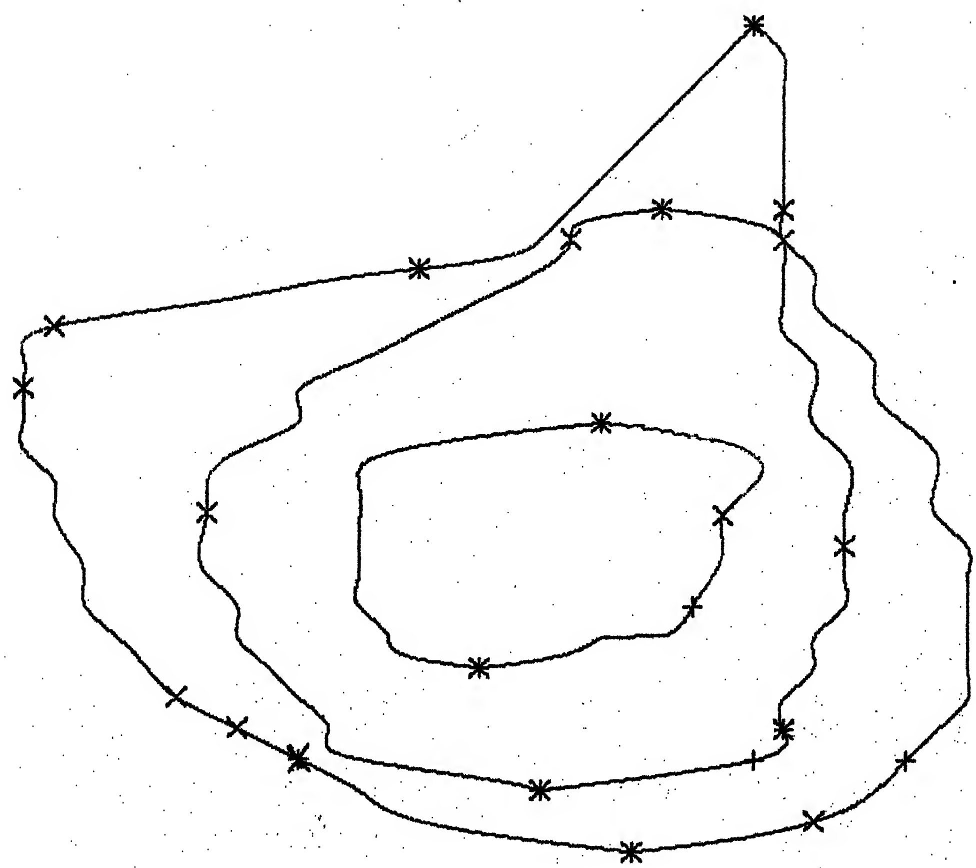
After going through all these results, it appears that the spline approximation is highly effective when applied to curves of length ranging from 10 to 40. For shorter curves, usually a proportionately high number of knots are required to tie up the various sections. For longer ones, the approximation procedure not only takes too much time but also ends up with a solution which is optimum only in the local sense. The routine takes a time ranging from 5 secs. to 12 secs. for approximating



EACH X DENOTES A SAMPLE OF THE DIGITISED MAP

ORIGINAL VERSION

FIGURE 4·1 CONTOUR MAP-1



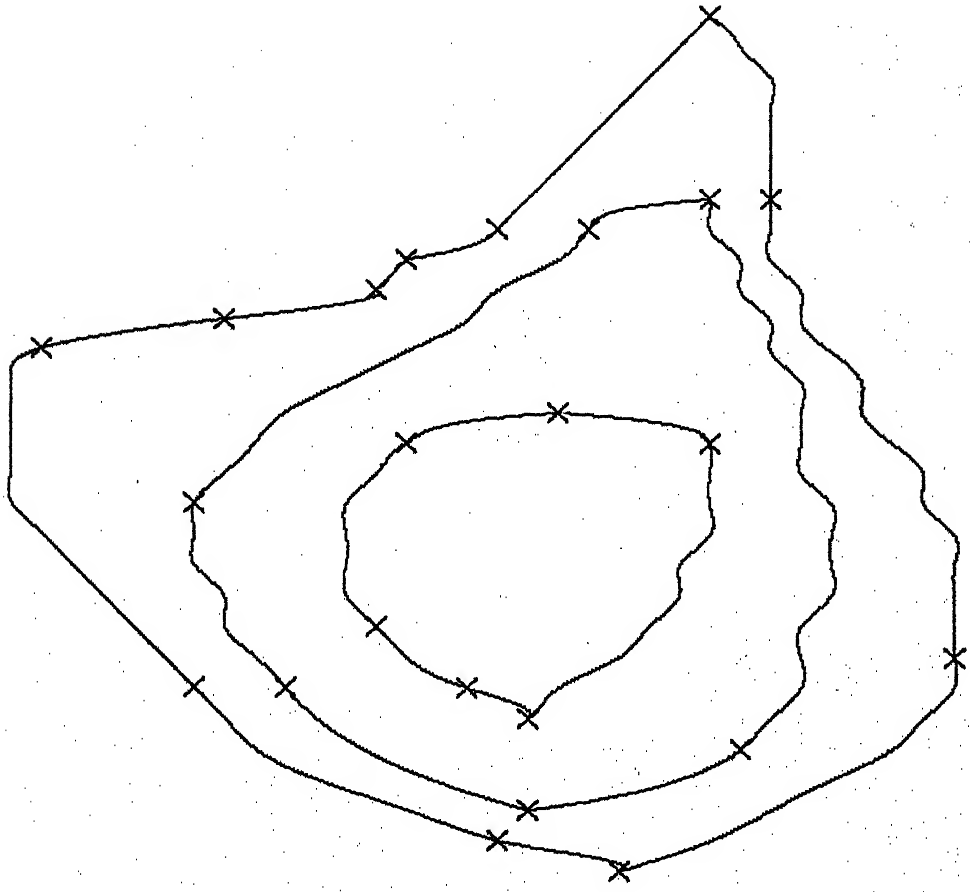
EACH + REPRESENTS A KNOT OF THE X-FIT

EACH X REPRESENTS A KNOT OF THE Y-FIT

APPROXIMATED VERSION-1

(PARAMETRIC APPROACH)

FIGURE 4.2 CONTOUR MAP-1

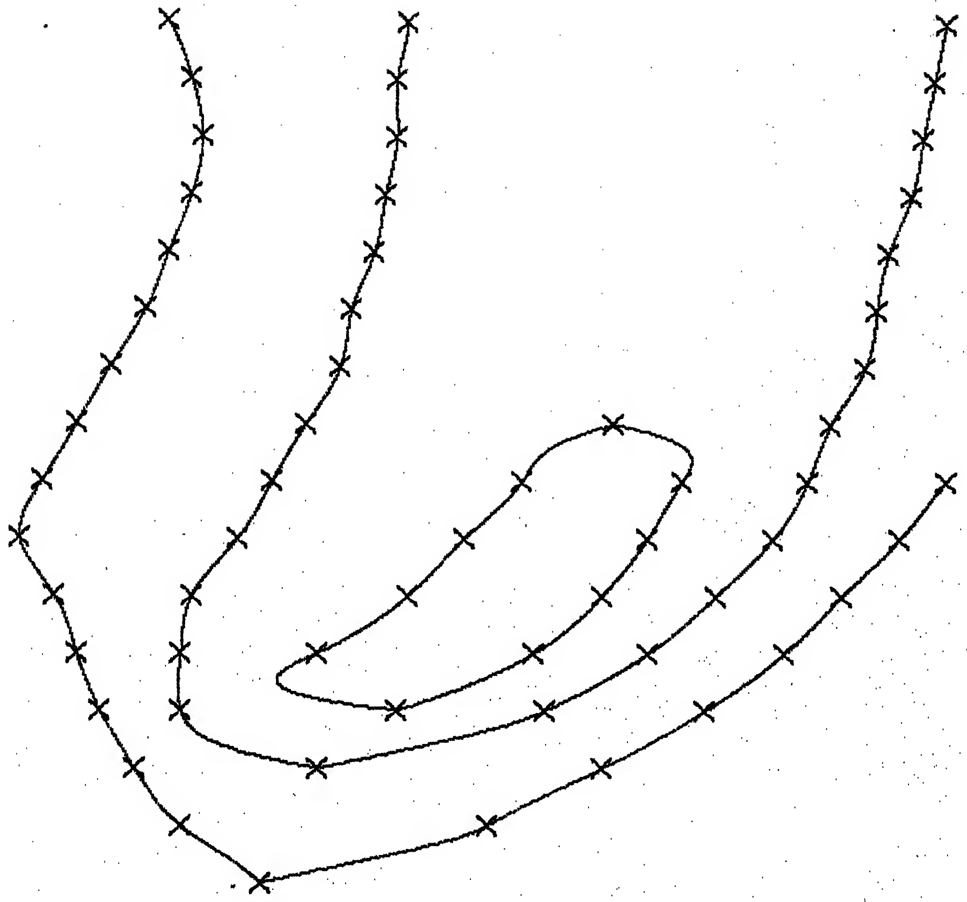


EACH X REPRESENTS A KNOT OF THE SPLINE-FIT

APPROXIMATED VERSION-2

(SEGMENTATION APPROACH)

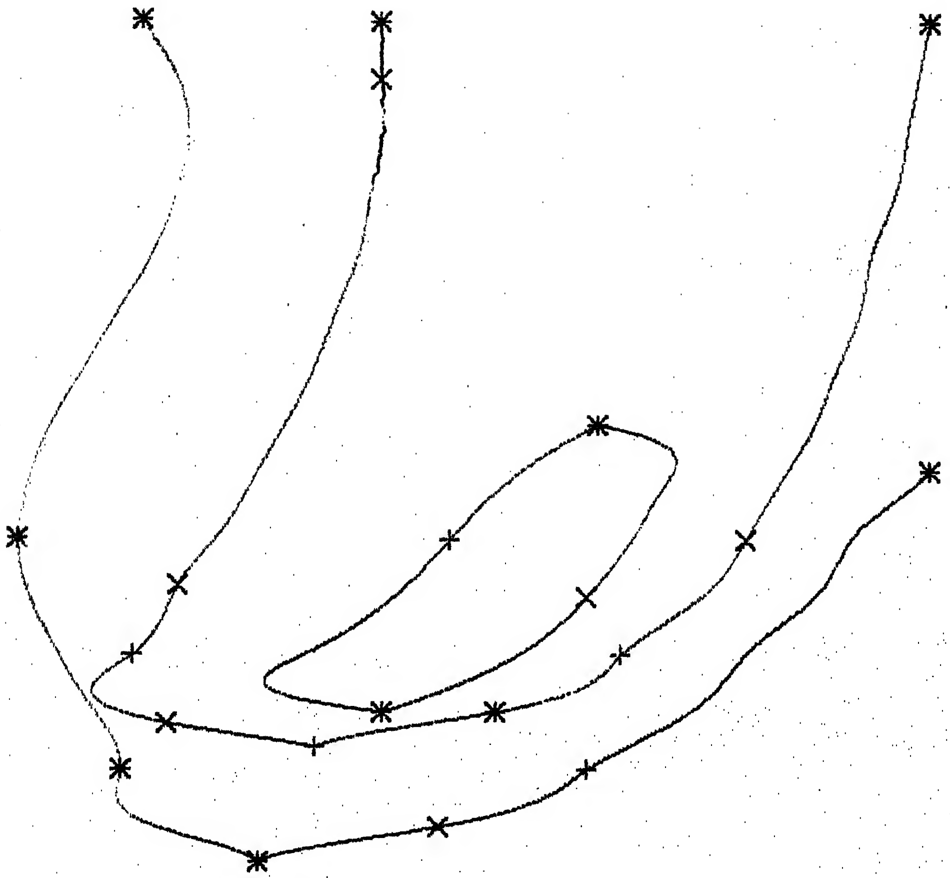
FIGURE 4.3 CONTOUR MAP-1



EACH X DENOTES A SAMPLE OF THE DIGITISED MAP

ORIGINAL VERSION

FIGURE 4.4 CONTOUR MAP-2



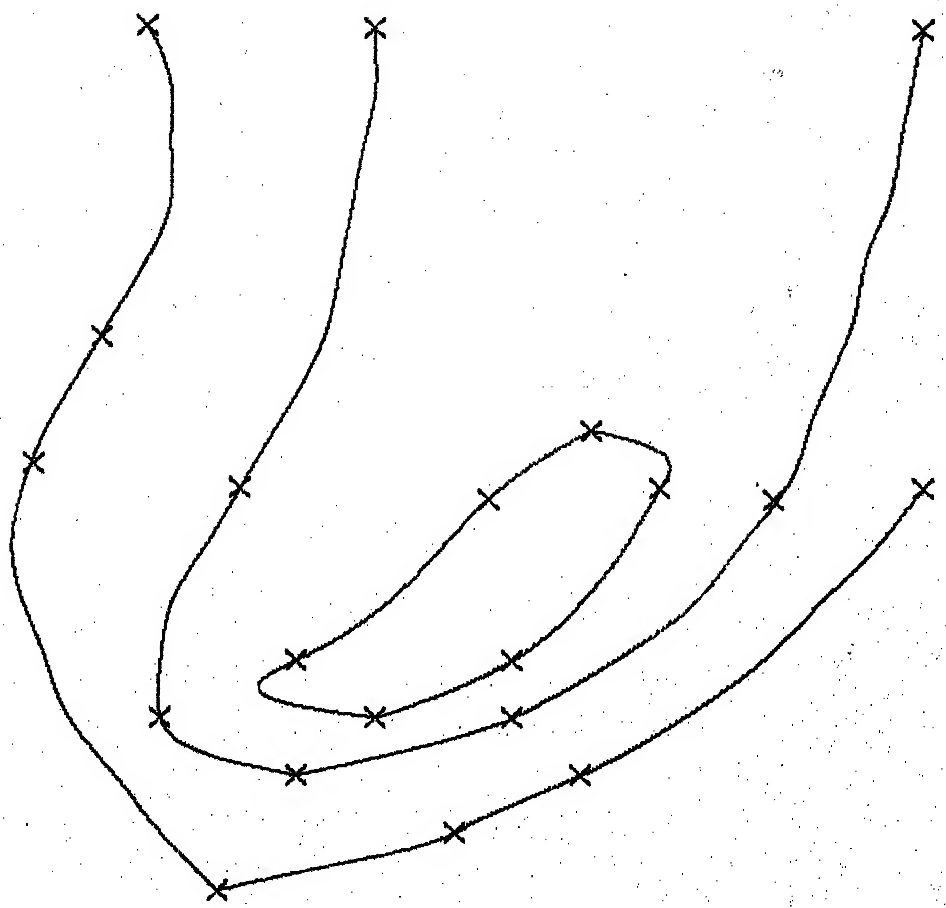
EACH + REPRESENTS A KNOT OF THE X-FIT

EACH X REPRESENTS A KNOT OF THE Y-FIT

APPROXIMATED VERSION-1

(PARAMETRIC APPROACH)

FIGURE 4.5 CONTOUR MAP-2



EACH X REPRESENTS A KNOT OF THE SPLINE-FIT

APPROXIMATED VERSION-2

(SEGMENTATION APPROACH)

FIGURE 4.6 CONTOUR MAP-2

TABLE 1

Parametric Approximation

			X vs. L			Y vs. L		
Figure	S.No.	Length	No. of knots	ERROR		No. of knots	ERROR	
				L_∞	L_2		L_∞	L_2
CONTOUR MAP 1	1	20	4	1.77	0.28	3	1.29	0.37
	2	40	5	1.00	0.12	9	1.20	0.25
	3	48	5	1.00	0.15	10	1.30	0.31
CONTOUR MAP 2	1	27	6	2.00	0.45	7	3.00	0.76
	2	23	6	2.00	0.44	6	3.00	0.87
	3	9	3	1.00	0.14	3	1.00	0.22

TABLE 2

Approximation after Segmentation

Figure	S.No.	Length	No. of knots	ERROR	
				L_∞	L_2
CONTOUR MAP 1	1	11	3	1.13	0.19
	2	9	3	0.50	0.18
	3	21	3	1.59	0.40
	4	16	3	1.08	0.30
	5	29	4	1.58	0.37
	6	19	5	1.00	0.51
CONTOUR MAP 2	1	14	4	1.17	0.30
	2	13	3	1.00	0.30
	3	16	4	2.00	0.70
	4	7	3	1.00	0.20
	5	6	3	0.10	0.10
	6	4	3	1.00	0.40

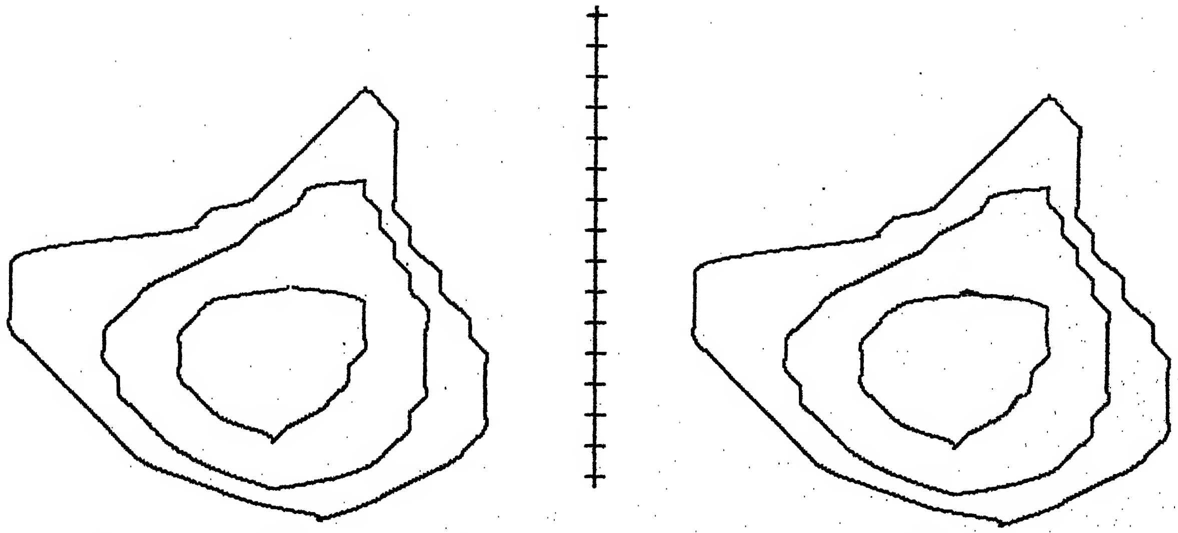


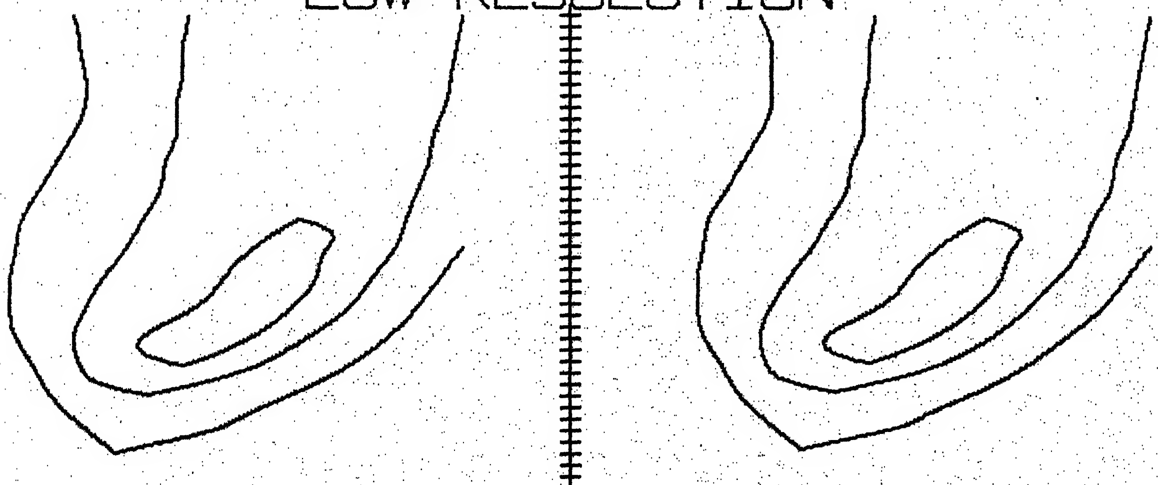
FIGURE 4.7 CONTOUR MAP-1

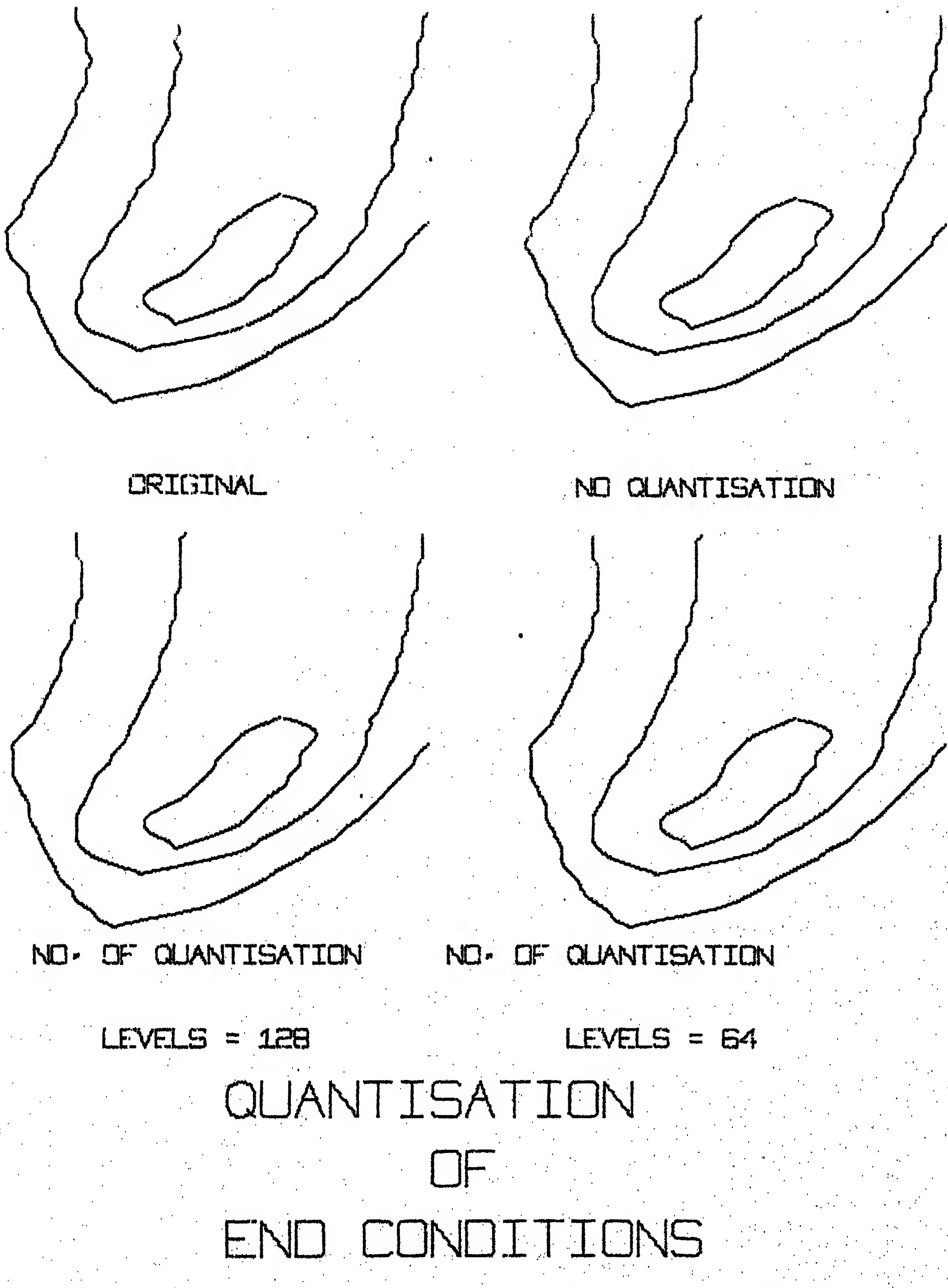
NO QUANTISATION

QUANTISATION OF
END CONDITIONS

NO. OF QUANTISATIO
LEVELS = 64

FIGURE 4.8 CONTOUR MAP-2
LOW RESOLUTION





QUANTISATION OF END CONDITIONS

FIGURE 4.9 CONTOUR MAP-2 HIGH RESOLUTION

TABLE 3

Effect of Quantization of End Derivatives

Figure	S.No.	Length	No.of knots	No. of quantization levels			
				No quant.	128	64	32
				L_∞ Error			
CONTOUR MAP 1	1	11	3	1	1	1	2
	2	9	3	1	1	1	1
	3	21	3	2	2	2	3
	4	16	3	1	1	1	1
	5	29	4	2	2	2	2
	6	19	5	1	2	3	4
CONTOUR MAP 2 LOW RESLN.	1	14	4	1	1	2	2
	2	13	3	1	1	1	1
	3	16	4	2	2	2	2
	4	7	3	1	1	1	1
	5	6	3	0	0	1	1
	6	4	3	1	5	8	13
CONTOUR MAP 2 HIGH RESLN.	1	36	3	2	2	2	2
	2	40	3	1	1	1	1
	3	16	3	1	1	1	1
	4	26	4	1	1	1	1
	5	31	3	1	1	1	1
	6	36	3	1	1	1	1
	7	25	3	1	1	1	1
	8	36	3	1	1	1	1
	9	30	4	1	1	1	1

a contour, made up of 15 sample points on IBM 7044. This routine is usually terminated when the approximation error is reduced to a value of about 1.0 in the minimax sense and a value of 0.3 in the least square sense. When the curve is unusually kinked, the minimax error is allowed upto the value of 2.0 and the least square error upto 0.9. On the average, this type of approximation reduces the number of points needed to represent the contour by a factor, close to 6:1. All figures, concerning the above subject, were drawn on the plotter attached to IBM 1800.

4.3 EVALUATION OF SYSTEM PERFORMANCE

Preliminary investigations on contour approximation favour the incorporation of the spline approximation scheme into the Synthetic Highs System. However, the system performance can be truly evaluated only after processing a picture signal.

4.3.1 I/O Structure of the Picture Signal:

The conversion of a picture into the digitised form accepted by the computer, is often achieved through a scanner equipment which converts the picture into the desired form and stores it on the tape. The processed picture is recorded back on the tape. The equipment is used once again to convert the digital figures into the normal pictorial form. Another form of computer output often accepted when such a facility is unavailable, is to

represent the various intensity levels by different combinations of alphabets. However, such a picture will be coarsely quantized. Besides this needs some of the lines to be overprinted on ones already printed. Owing to lack of access to either of these facilities, only a two level picture was processed. Since the main aim is to test the suitability of the contour approximation technique with relevance to picture transmission, it is hoped that the results obtained would not differ significantly from those using a multilevel picture. Even though the input signal itself consists of only two levels, the system is assumed to be capable of transmitting many more levels so that the "Lows" of the picture are received without significant loss of accuracy.

4.3.2 Coding the Contour Information:

Although the spline approximation reduces the number of points to be transmitted, it does so, as the cost of removing the neighbourhood relation between adjacent samples transmitted. Whereas in Graham's scheme, for every additional point to be transmitted only 3 more bits have to be used, in this scheme a lot more than three is required. This calls for stringent limits to be imposed on the number of bits required to transmit the contours. This problem can be tackled in the following way:

In each contour, only the starting knot is transmitted with full details pertaining to its position. For

the subsequent knots only their relative positions with reference to preceding knots are transmitted. Moreover, since the distance between the adjacent knots will vary from contour to contour, a code word indicating the number of bits used for specifying these knots could be transmitted. This way, some gain can be made on the number of bits transmitted. However, this is done at the cost of redundancy and therefore sufficient error correcting capacity should be imparted to the code word indicating the number of bits representing the knots. Further, there may be some short contours in the picture which can not be efficiently approximated. It is advisable to block such contours into one group and transmit them separately in the normal manner of contour coding. This will also improve the coding efficiency of the entire scheme.

The picture processed by the simulated system is shown in Figure 4.10 and the results of approximation tabulated in Table 4 and Figures 4.11 to 4.14. The procedure for calculating the required number of bits for transmitting the given picture is clearly indicated in Appendix IV. The final results indicate that a reduction of high frequency information by a factor of about 6 to 40 is possible with this system, i.e., it can provide an improvement by a factor of approximately 1.7 over Graham's scheme. Since 'Lows' need only a small fraction of the total number of bits used, these figures hold good even for

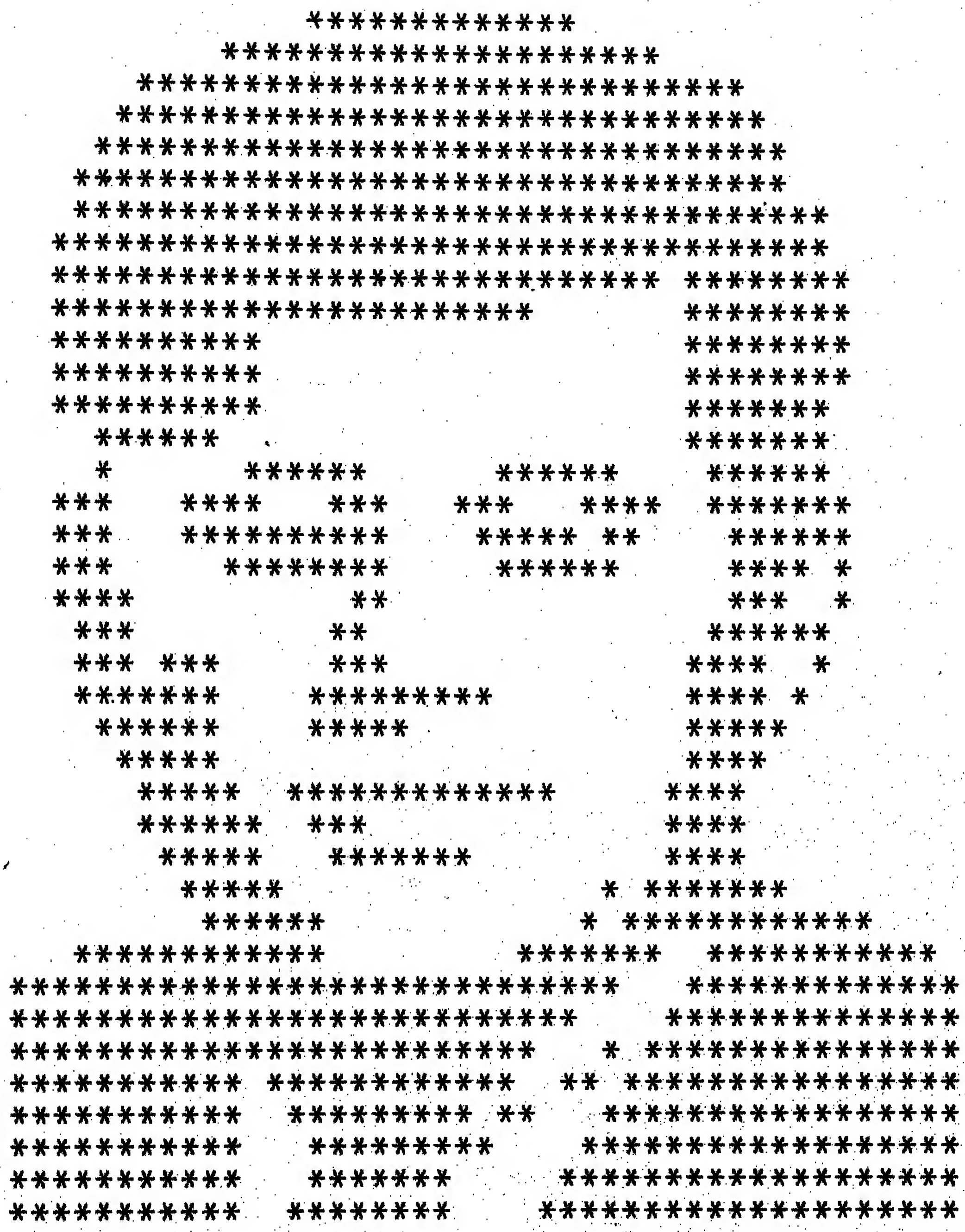


FIGURE 4-10 ORIGINAL PICTURE

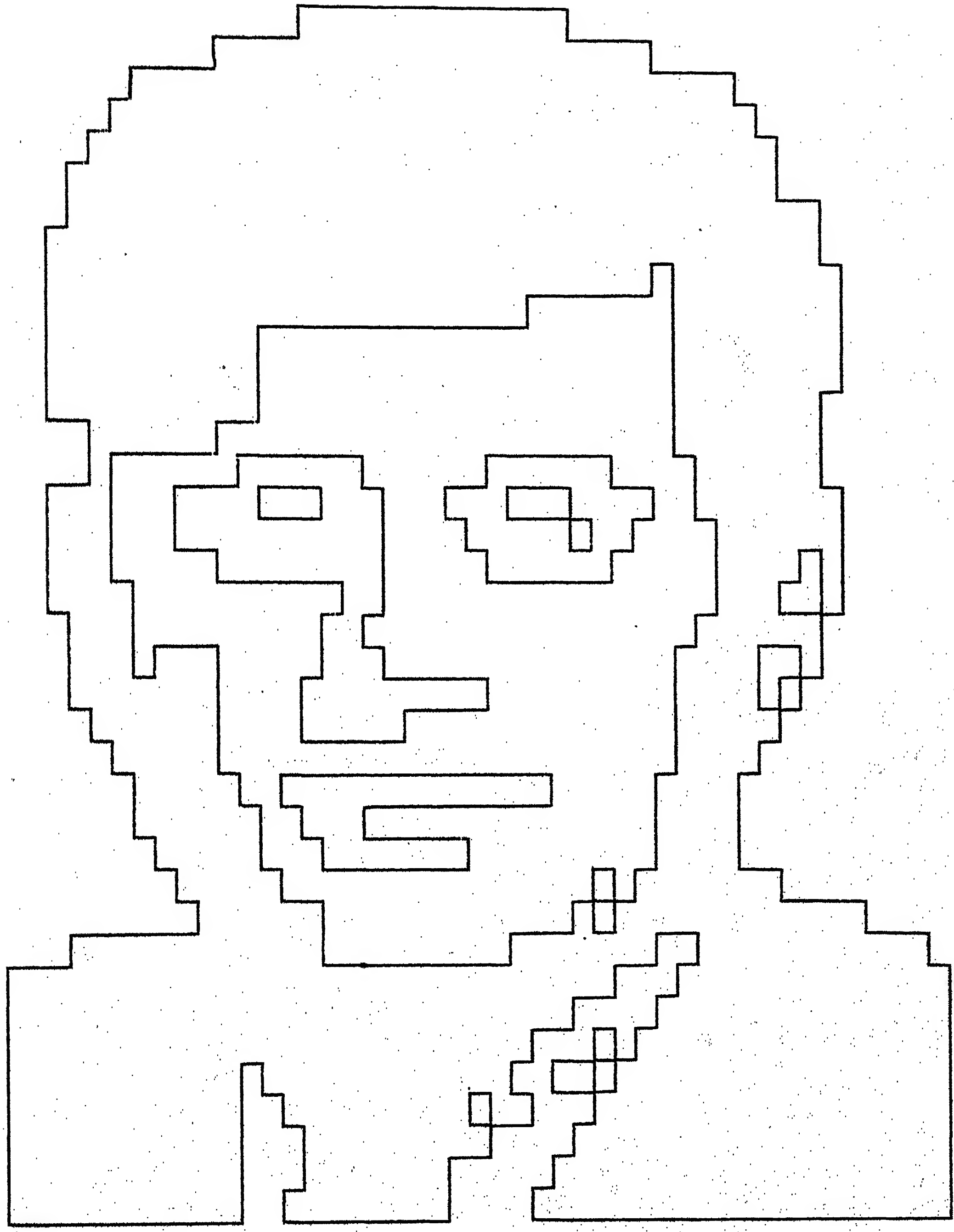


FIGURE 4·11 CONTOURS OF THE PICT

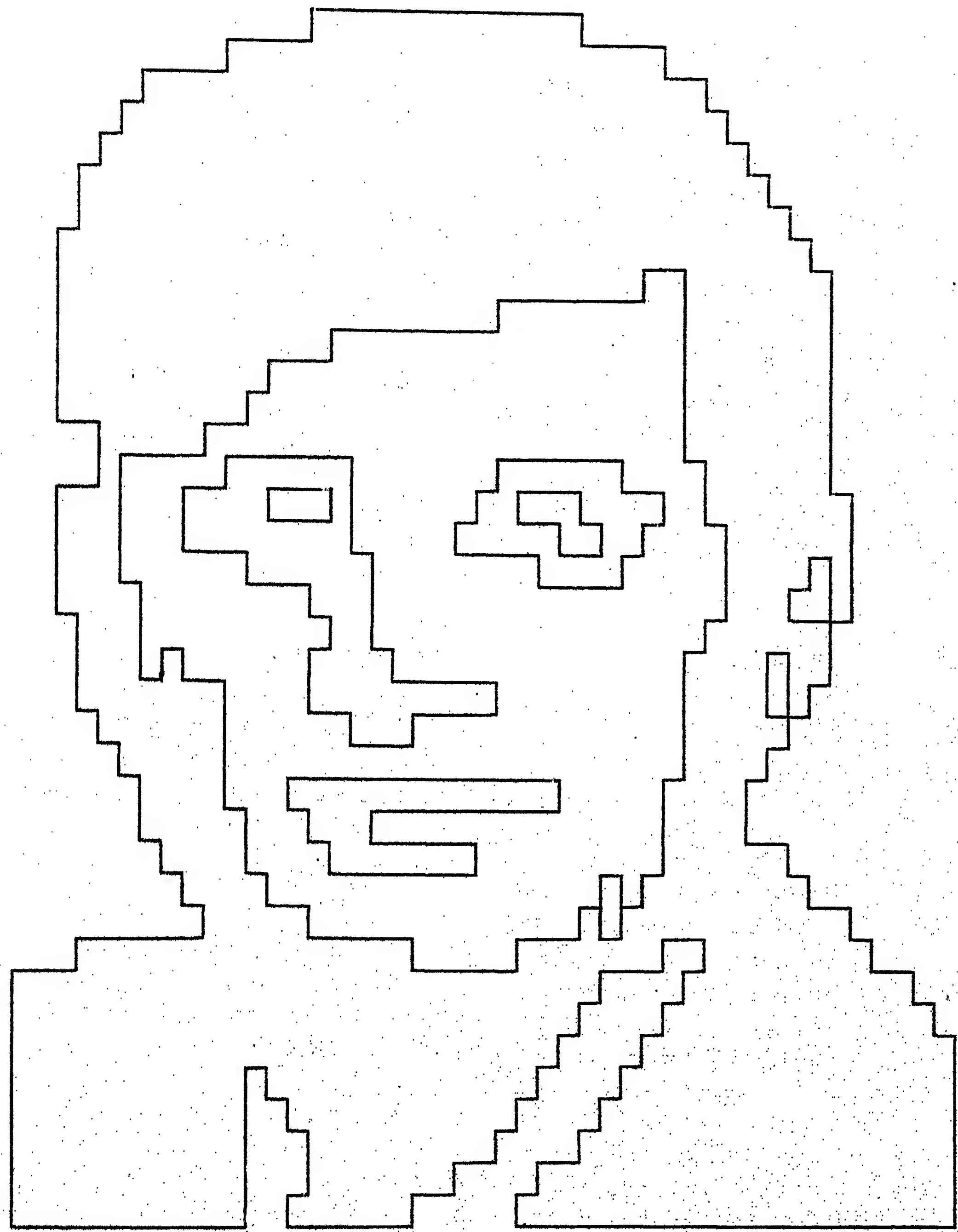


FIGURE 4-12 RECONSTRUCTED CONTOUR

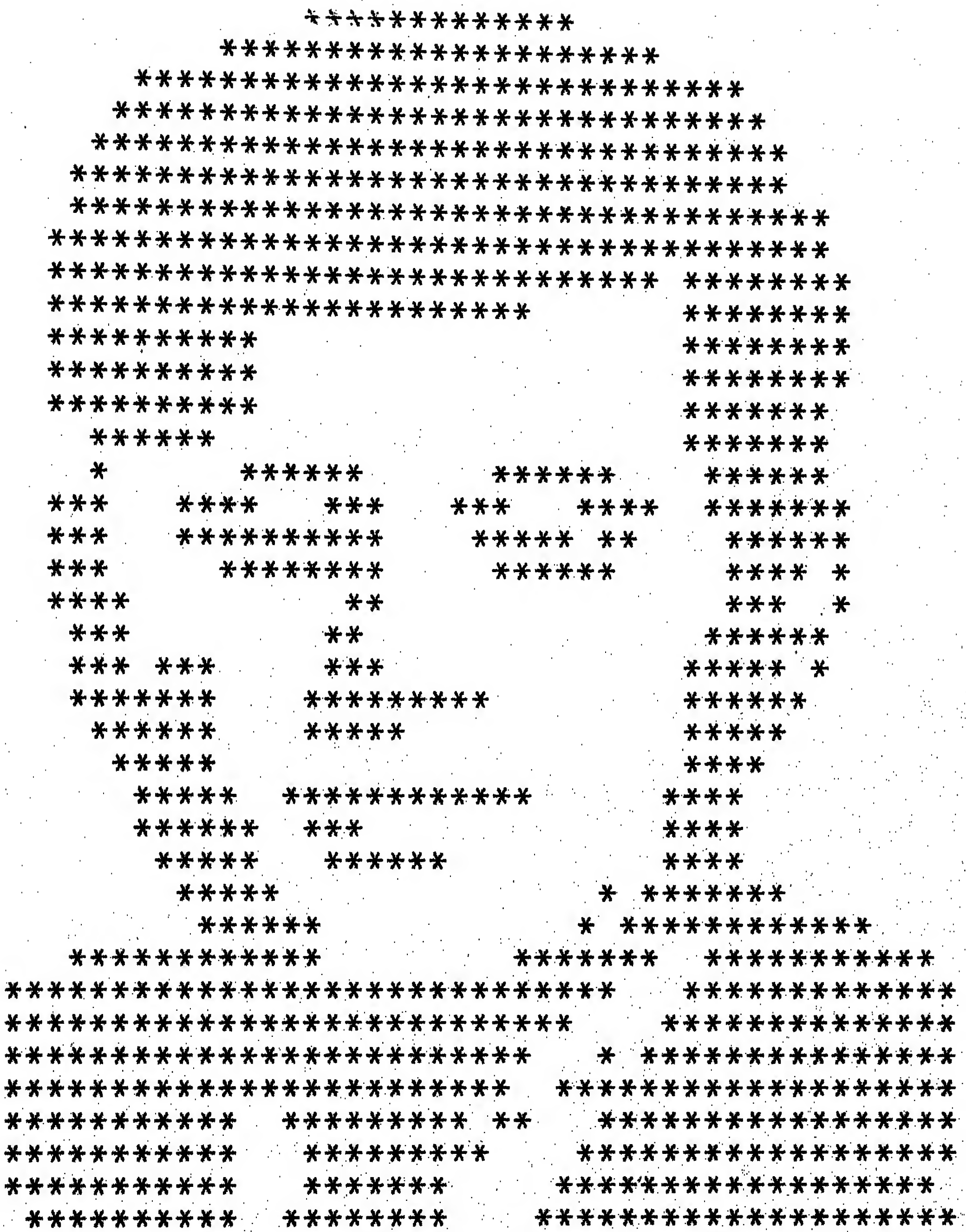


FIGURE 4-13 RECEIVED PICTURE
GRAHAM'S SCHEME

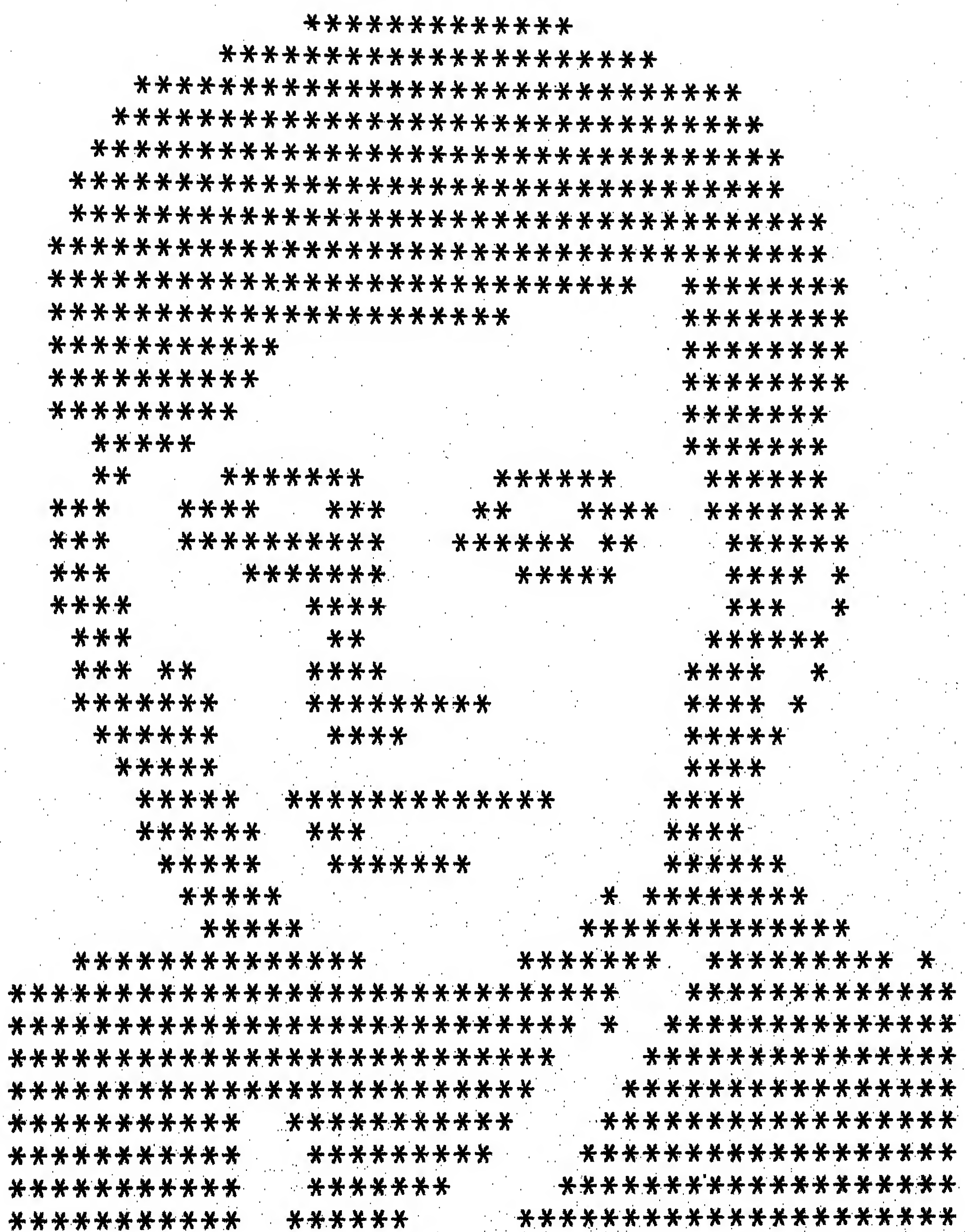


FIGURE 4-14 RECEIVED PICTURE
PROPOSED SCHEME

Table 4

Approximation of Picture Contours

S.No.	No. of samples in the contour		No. of knots	Bits/additional knot	Total no. of bits required
	Before smoothing	After smoothing			
1	9	8	3	3	36
2	30	7	3	5	44
3	24	16	3	3	36
4	31	16	3	4	40
5	28	9	4	4	48
6	28	11	4	5	54
7	27	6	3	4	40
8	37	9	4	6	60
9	26	9	4	3	42
10	7	7	3	3	36
11	35	20	5	3	48
12	46	19	5	5	64
	TOTAL NUMBER OF BITS =				548

the signal in entire frequey domain. Thus the proposed system can provide a reduction in bandwidth by a factor varying from 6 to 40 in comparison with 6 bit PCM scheme.

Chapter V

CONCLUSIONS

This system has definite improvements to its credit which can be favourably implemented in Graham's scheme. One important gain made in the coding is achieved by generating the gradient direction from the contour orientation. This sharpens the picture by thinning its contours which have already been smoothened by spline approximation. These effects have not been observed on the results obtained, probably due to the fact that only low resolution pictures, that too quantized to two levels, have been used.

This system suffers from the drawback common to all Synthetic Highs Systems. Any picture not only consists of contours defined by changes in gray level, but includes a few edges characterised by the statistical properties of regions lying on either side. Although there are algorithms available for extraction of such contours, there is no known methods of regenerating signals providing the textural properties. Schreiber[15] has indicated as to how to attack this problem. The gain in the bandwidth obtained using splines can very well be utilised to accommodate such textural information.

Since the system has not yet passed the crucial test of transmitting multi-level pictures, any further elucidation on the subject is likely to be treated as a vain boast. As such, a follow-up using multi-level pictures is deemed essential.

REFERENCES

1. Ahlberg, J.H., Nalson, E.N. and Walsh, J.L., The Theory of Splines and Their Applications, Academic Press, New York, 1967.

2. Barradole, I. and Young A, "A note on numerical procedures for approximation by spline functions", Computer Journal, Vol.9, pp. 318-320, 1967.

3. De Boor, C. and Rice, J.R., "Least squares cubic spline approximation I - Fixed knots", CSD TR-20, Purdue University, U.S.A., April 1968.

4. De Boor, C. and Rice, J.R., "Least squares cubic spline approximation II - Variable knots", CSD TR-21, Purdue University, U.S.A., April 1968.

5. Esch, R.E. and Eastman, W.L., "Computational methods for best spline approximation", Journal of Approximation Theory 2, pp. 85-96, 1969.

6. Graham, D.N., "Image transmission by two dimensional contour coding", Proc. IEEE, Vol.55, No.3, pp.336-346, March 1967.

7. Huang, T.S. and Tretiak, O.J., "Research in Picture processing", Chapter 3 in Optical and Electro-optical Information Processing, M.I.T. Press, 1965.

8. Lawson, C.L.,"Segmented rational minimax approximation - Characteristic properties and computational methods", J.P.L. Tech. Rep. 32-579, Passadena, California, U.S.A., 1963.

9. Pan, J.W., "Picture Processing", Q.P.R. No.61, Research Lab. of Electronics, M.I.T., p.229, July 15, 1962.

10. Pingle, K.K., "Visual perception by a computer" in Automatic Interpretation and Classification of Images, Academic Press, New York, pp. 277-284, 1969.

11. Powell, M.J.D., "On best L_2 spline approximations", AERE Report, T.P. 264, Harwell, England, 1966.

12. Rice, J.R., "Spline functions", Chapter 10 in The Approximation of Functions, Vol.2, pp. 123-167, Addison-Wesley, 1968.

13. Rosenfeld, A., Thomas, R.B. and Lee, Y.H., "Edge and curve enhancement in digital pictures", University of Maryland Computer Center Technical Report 69-93, May 1969.

14. Sakal, T., Nagao, M. and Fujibayashi, S, "Line extraction and pattern detection in a photograph", Pattern Recognition 1, pp. 233-248, March 1969.

15. Schreiber, W.F., "Picture Coding", Proc. IEEE, Vol.55, No.3, pp. 320-330, March 1967.

16. Schreiber, W.F., Knapp, C.F. and Kay, N.D., "Synthetic Highs - an Experimental TV Bandwidth Reduction System", J.SMPTE, Vol.68, pp. 523-537, August 1959.

17. Schreiber, W.F., "The mathematical foundation of synthetic Highs systems", Q.P.R.No.68, RLE, M.I.T., p.140, January 15, 1963.

18. Schumaker, L.L. "On computing best spline approximations", M.R.C. Report No.833, Madison, Wisconsin, U.S.A., 1968.

Appendix I

ON PERFORMANCE OF THE RECEIVER TO NOISY INPUT

The analysis is confined to signals of single dimension, since it can be extended to higher dimensions without any difficulty. The receiver in Synthetic Highs System consists of a gradient regenerator followed by a high pass filter.

Let $H(w)$ be the response of the HPF. Then

$$H(w) = \frac{1 - L(w)}{jw} \qquad \text{(AI.1)}$$

where $L(w)$ is the filter response of the LPF in the transmitter.

Let $N_i(w)$ be the spectral density of noise input to the filter and $N_o(w)$ the corresponding output density. Then

$$N_o(w) = \left| \frac{1 - L(w)}{jw} \right|^2 N_i(w) \qquad \text{(AI.2)}$$

Total noise output power is given by

$$S(N_o) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{1 - L(w)}{jw}^{2} N_i(w)\, dw$$

Let us assume that $N_i(w) = N_i$, i.e., $N_i(w)$ is white. Then

$$S(N_o) = \frac{N_i}{2\pi} \int_{-\infty}^{+\infty} \frac{|1 - L(w)|^2}{w^2}\, dw \qquad \text{(AI.3)}$$

$N_o(w)$ is unbounded at $w = 0$ unless

$$\lim_{w \to 0} \left| \frac{1 - L(w)}{w} \right| < \infty \qquad \text{(AI.4)}$$

The necessary condition for (AI.4) to hold is

$$\lim_{w \to 0} L(w) = 1 \qquad \text{(AI.5)}$$

If $l(t)$ is the time response of the LPF

$$L(w) = \int_{-\infty}^{+\infty} l(t) \exp(-jwt) dt$$

Therefore,

$$L(w)\Big|_{w=0} = \int_{-\infty}^{+\infty} l(t)\, dt$$

Applying condition (AI.5)

$$\int_{-\infty}^{+\infty} l(t)\, dt = 1 \qquad \text{(AI.6)}$$

i.e., the filter response should be normalized so that condition (AI.6) is satisfied.

In the absence of information about "Lows"

$$H(jw) = \frac{1}{jw}$$

From (AI.2) it follows that under such conditions, any small error in gradient regeneration or noise from any other source causes enormous disturbance, particularly, in the low frequency regions. Presence of LPF in the transmitter helps to reduce such noise generation.

Appendix II

SOME QUALITATIVE RESULTS
CONNECTING
CONTOUR THRESHOLD AND LOW PASS CUT-OFF

The following analysis although not rigorous and may, in fact, contain quite a few loopholes, tries to bring out the close connection between the threshold set up for contour extraction and cut-off frequency of the filter transmitting the "Lows". The analysis is done on a digitised signal of single dimension and this can be easily extended to cases of higher dimensions.

Let $x(t)$ be the time limited signal sampled at instants $t_1, t_2, \ldots, t_n$. Then

$$x(t) = \sum_i a_i \, u(t-t_i) \qquad \text{(AII.1)}$$

where $u(t-t_i)$ is the unit step function with discontinuity at t_i and

$$a_i = x(t_i) - x(t_{i-1})$$

for all $i = 1, \ldots, n$ and $x(t_o) = 0$

Let $x^*(t)$ represent the signal $\sum_i a_i^* \; u(t-t_i)$

where $a_i^* = a_i$ whenever $a_i <$ threshold

$\qquad = 0$ otherwise

i.e., $x^*(t)$ represents the information left out after transmitting all the contour points.

Taking the Laplace transform

$$X^*(s) = \sum_i \frac{a_i^*}{s} \exp(-st_i) \qquad \text{(AII.2)}$$

Substituting $s = jw$ in (AII.2) we see that

$$\left|X^*(jw)\right| \leqslant \sum_i \frac{a_i^*}{jw} \exp(-jwt_i)$$

$$\leqslant \sum_i \frac{a_i^*}{w} \qquad \text{(AII.3)}$$

Thus the spectrum of the left out portion of the signal has an upper bound given by (AII.3).

Since the purpose of the "Lows" is to supplement the information provided by the contours, "Lows" should contain, as much as possible information relating to $x^*(t)$.

If w_c is the cut-off frequency of LPF and $e(jw)$ represents that portion of $X^*(jw)$ not contained in "Lows" either,

$$\begin{aligned} e(jw) &= X^*(jw) \quad ; \quad |w| \geqslant w_c \\ &= 0 \qquad\qquad\quad \text{otherwise} \end{aligned}$$

Total power E_o contained in $e(jw)$ is given by

$$E_o = \frac{1}{2\pi} \int_{|w| \geqslant w_c} e(jw)\, dw$$

$$\frac{1}{2\pi} \int_{|w| \geqslant w_c} \left(\sum_i a_i^*\right)/w \; dw \qquad \text{(AII.4)}$$

Let w_{max} represent the maximum frequency content of the signal x(t), then

$$E_o \leq \left(\left(\sum_i a_i^*\right)/2\pi \right) \int_{w_c}^{w_{max}} \frac{dw}{w}$$

$$\leq \left(\left(\sum_i a_i^*\right)/2\pi \right) \ln \frac{w_{max}}{w_c} \qquad \text{(AII.5)}$$

$\sum_i a_i^*$ is directly related to the threshold in the sense that when the threshold is increased, $\sum_i a_i^*$ also increases. From (AII.5) it is evident that an obvious way to counter the increase in E_o is to increase w_c. Thereafter, the interdependence of these two quantities automatically follows.

Appendix III

SYNTHESIS OF RECONSTRUCTION FILTER

Let $h(x,y)$ be the response of the reconstruction filter and $b(x,y)$ the image signal. The entire system barring the thresholding operation, is represented by the equation

$$(\nabla b) * \bar{h} + b * l = b \qquad \text{(AIII.1)}$$

where * indicates a dot product convolution. The Fourier transform of this equation yields

$$(jw_x H_x + jw_y H_y) B + BL = B$$

where L is the frequency response of the low pass filter. For frequencies at which $B \neq 0$

$$jw_x H_x + jw_y H_y = 1-L$$

which when inverse-transformed yields

$$\nabla . \bar{h} = u_o(x,y) - l(x,y) \qquad \text{(AIII.2)}$$

where $u_o(x,y)$ is the Dirac delta function in two dimensions. If the filter l is circularly symmetric, (AIII.2) reduces to the form

$$\nabla . \bar{h}(r) = u_o(r) - l(r) \qquad \text{(AIII.3)}$$

The corresponding difference equation is given by

$$h_x(i,j) - h_x(i,j-1) + h_y(i,j) - h_y(i-1,j)$$

$$= \delta_{ij} - l(i,j) \qquad \text{(AIII.4)}$$

where $\delta_{ij} = 1 \qquad i = j = 1$

$\qquad = 0 \qquad$ otherwise

If the LPF is circularly symmetric,

$$h_x(i,j) = h_y(j,i)$$

Then eqn.(AIII.4) becomes

$$h_x(i,j) - h_x(i,j-1) + h_x(j,i) - h_x(j,i-1)$$

$$= \delta_{ij} - l(i,j) \qquad \text{(AIII.5)}$$

Eqn.(AIII.4) must hold good for all i and j even though

$$l(i,j) = 0 \quad \text{for} \quad i,j > n$$

But it is desirable to have h_x zero outside the spatial extent of LPF so that

$$h_x(i,j) = 0 \quad i,j > n$$

It can be shown that in such a situation

$$h_x(i,n) = 0$$
$$h_x(1,1) = (1 - l(1,1))/4 \qquad \text{(AIII.6)}$$

Setting

$$h_x(i,j) = 0 \qquad i > j+1 \qquad \text{(AIII.7)}$$

We can get the following equations:

$$
\begin{aligned}
h_x(i+1,i) &= (l(i+1,i+1) + 2h_x(i+1,i+1))/2 \\
h_x(i,i) &= l(i, i+1) + h_x(i,i+1) + h_x(i+1,i) \\
h_x(k,i) &= l(k,i+1) + h_x(k,i+1) \qquad 1 \leq k \leq i-1 \\
h_x(1,1) &= (1-l(1,1))/4
\end{aligned}
\qquad \text{(AIII.8)}
$$

A detailed description of this reconstruction, aided by an example, is given in Graham's paper[6].

Appendix IV

CALCULATION OF BANDWIDTH REDUCTION RATIO

The strategy adopted for computing the bandwidth requirements of the proposed scheme is briefly described below:

The picture shown in Figure 4.10 is processed using the proposed scheme as well as Graham's scheme. The bandwidth requirements of both the schemes are computed using the results obtained. Assuming that the ratio of the two bandwidth requirements will not change appreciably while processing multi-level pictures, the bandwidth requirements of the proposed scheme are deduced from the results obtained by Graham[6]. The computations leading to the final result are indicated below.

Graham's Scheme:

No. of bits required for --

(a) Starting point of the contour (for 64x64 picture) = 12

(b) Position information of each additional contour point = 3

(c) Gradient direction at each point = 3

(d) Gradient magnitude for each contour = 3

No.	Description of features	Number present	Bit requirement
1	Starting point	18	18 x 12 = 216
2	Additional contour point	352	352 x 3 =1056
3	Gradient magnitude		6 x 18 = 108
4	Gradient direction		352 x 3 =1056
	Total Bits =		2436

The Proposed Scheme:

Referring to Table 4, all the approximated contours put together need only 548 bits. Let us see how this figure has been arrived at.

Considering the 5th entry of Table 4,

No. of knots = 4

Bits/additional knot = 4

No. of bits for starting point = 12

end derivatives= 12

3 addl. knots = 24

Total Bits = 48

This does not take into consideration the no. of bits required for code word specifying the bit configuration of each contour. Assuming that each such code word requires 6 bits, an additional 72 bits are required for 12 contours. In addition 36 bits are required for transmitting the gradient magnitude.

There are 202 more contour points in 25 contours to be transmitted by direct contour coding using the procedure described in Chapter III, which requires only 2 bits of each additional node. These points need about 729 bits for transmission. Another 50 bits are allotted for delimiting code words separating various sections of the coder output. Altogether the proposed scheme requires

$$548 + 729 + 36 + 72 + 50 = 1435 \text{ bits.}$$

The improvement factor of the proposed scheme over Graham's scheme = 2436/1435 = 1.7.

Since Graham's scheme shows an improvement over 6 bit PCM by a factor varying from 4 to 23, this scheme is likely to give a reduction ratio ranging from 6 to 40 when compared to 6 bit PCM. From the results obtained by Graham, the low pass signal needs only 1/10th of the entire bandwidth and hence it can be safely assumed that the same figures hold good for the entire signal.

As seen from the results, the approximated contours representing well over 400 points need only 656 bits as compared to the 729 bits for 202 remaining points. The importance of approximation techniques in this application is glaringly obvious from the above figures.

APPENDIX V

SIMULATION PROGRAMS

```
C              *******************************************
C              *
C              * SIMULATION OF LOW PASS FILTER           *
C              *                                         *
C              *******************************************
$IBFTC *MAIN*
C
C    THE PICTURE INTENSITY MATRIX IS READ INTO ARRAY-PIC
C    THE OUTPUT OF THE FILTER IS STORED IN    PICLOW
C    FIL1(I)*FIL2(J)  GIVES THE APPROPRIATE FILTER SAMPLE
C    ARRAY ALIM USED  FOR PICTORIAL OUTPUT OF THE PICTURE
C    SIGMA DETERMINES THE  CUT OFF FREQUENCY OF THE FILTER
      COMMON/FIL1/FIL1(25),FIL2(25),PIC(50,65),PICLOW(50,65)
       DIMENSION FILL(50)
      INTEGER PIC,PICLOW,FIL1,FIL2
      READ 1,NX,NY
    1 FORMAT(2I3)
      READ 2,((PIC(I,J),J=1,NY),I=1,NX)
    2 FORMAT(40I2)
      SIGMA=3.
      PRINT 111
  111 FORMAT(*1 ORIGINAL PICTURE*/)
      CALL OUTPUT(0)
      DO 3 I=1,NX
      DO 3 J=1,NY
    3 PICLOW(I,J)=0.
      SUM=0.
C
C  FILTER SAMPLES-GENERATION THROUGH THEIR KERNELS
C
      DO 4 I=1,10
      R=(I-1)*(I-1)
      R=R/(2.*SIGMA)
      FILL(I)=1./EXP(R)
    4 SUM=SUM+FILL(I)
      SUM=2.*SUM-1.
      DO 5 I=1,1?
      FIL1(I)=FILL(I)*1.E03/SUM
    5 FIL2(I)=FIL1(I)
   12 CALL LOWFIL(NX,NY)
      DO 61 I=1,NX
      DO 61 J=1,NY
   61 PICLOW(I,J)=PICLOW(I,J)/1000000
   51  FORMAT(X,20I6)
```

```
C
C    OUTPUTTING OF LOW PASS SIGNAL
C
        PRINT 51,((PICLOW(I,J),J=1,NY),I=1,NX)
        PRINT 101,SIGMA
        CALL OUTPUT(1)
  101 FORMAT(*1 LOW PASS VERSION OF THE PICTURE*/
     1* SIGMA = *,F12.6/)
        PUNCH 5 ,((PICLOW(I,J),J=1,NY),I=1,NX)
   50 FORMAT(40I2)
        ISW=ISW+1
        PUNCH 53
   53   FORMAT(20X,4HOVER)
  500 STOP
      END
*IBFTC LOWFIL
C
C    THIS ROUTINE PERFORMS THE NECESSERY CONVOLUTION
C    NX , NY   - SIZE OF THE PICTURE ARRAY
C
      SUBROUTINE LOWFIL(NX,NY)
      COMMON /FILT/FIL1(25),FIL2(25),PIC(50,65),PICLOW(50,65)
      INTEGER PIC,PICLOW,FIL1,FIL2
      INTEGER ADD1
      N=10
      DO 20 I=1,NX
      DO 20 J=1,NY
      N1=I
      IF(I.GT.N) N1=N
      N2=J
      IF(J.GT.N) N2=N
      DO 21 K=1,N1
      I1=I-K+1
      ADD1=0
      DO 25  L=1,N2
      J1=J-L+1
   25  ADD1=ADD1+FIL1(   L)*PIC(I1,J1)
      N3=NY-J
      IF(N3.GE.N)  N3=N-1
      IF(N3.LE.0) GO TO 21
      DO 27 L=1,N3
      L1=L+1
      L2=J+L
   27 ADD1=ADD1+FIL1(L1)*PIC(I1,L2)
   21 PICLOW(I,J)=PICLOW(I,J)+FIL2(K)*ADD1
      N4=NX-I
      IF(N4.GE.N)  N4=N-1
      IF(N4.LE.0) GO TO 20
      DO 22 K=1,N4
      K1=K+1
      K2=I+K
       ADD1=0
      DO 23 L=1,N2
       J1=J-L+1
   23 ADD1=ADD1+FIL1(L)*PIC(K2,J1)
```

```
      IF(N3.LE.0) GO TO 22
      DO 24 L=1,N3
      L1=L+1
      L2=J+L
   24  ADD1=ADD1+FIL1(L1)*PIC(K2,L2)
   22  PICLOW(I,J)=PICLOW(I,J)+FIL2(K1)*ADD1
   20 CONTINUE
      RETURN
      END
```

```
C            ***************************************
C            *                                     *
C            *  CONTOUR EXTRACTION  MODIFIED SCHEME *
C            *                                     *
C            ***************************************
C
C  THE INPUT PICTURE IS READ INTO ARRAY   PIC
C  THE CONTOURS EXTRACTED FROM THE PICTURE ARE STORED IN X AND Y
C  Z     CONTAINS THE GRADIENT ACROSS THESE EDGES
C
      COMMON/PIC1/PIC(50,65)
      COMMON/CONT1/MADDON(50,65),X(100),Y(100)
      DIMENSION ALIN(62)
      INTEGER X,Y,CCT,CST
       COMMON/PAR1/IDIF
      INTEGER PIC
      DATA BLANK,DOT/1H ,1H./
      READ 1,NX,NY
    1 FORMAT(2I3)
      DO 5 I=1,NX
      DO 5 J=1,NY
      MADDON(I+1,J+1)=0
      PIC(I+1,J+1)=0
    5 MADDON(I,J)=0
      PIC(1,NY+1)=0
      PIC(NX+1,1)=0
      MADDON(1,NY+1)=0
      MADDON(NX+1,1)=0
      READ 2,((PIC(I,J),J=1,NY),I=1,NX)
    2 FORMAT(40I2)
      IDIF=1
      CALL CONTO(NX,NY,7,7)
   32 CONTINUE
   25 PRINT 9
    9 FORMAT(*1 CONTOURS OF
```

```
      DO 8 I=1,NX
      DO 6 J=1,NY
      ALIN(J)=BLANK
    6 IF(PIC(I,J).EQ.1) ALIN(J)=DOT
      PRINT 10,(ALIN(J),J=1,NY)
C 10  FORMAT(1H ,20X,55A2)
   10 FORMAT(1H ,20X,65A1)
    8 CONTINUE
      STOP
      END
$IBFTC CONTO
C
C   IST  - CONTOUR START  THRESHOLD
C   ICT - CONTOUR CONTINUATION THRESHOLD
C   NX,NY   -SIZE OF THE PICTURE INPUT ARRAY
C
      SUBROUTINE CONTO(NX,NY,ICT,IST)
      COMMON/PIC1/PIC(50,65)
      COMMON/CONT1/MADDON(50,65),X(100),Y(100)
      INTEGER X,Y,CCT,CST
       INTEGER Z(100)
      INTEGER PIC
      COMMON/FAP/IDNI,IDNS
      LX=100
      CCT=ICT
      CST=IST
      NX1=NX+1
      NY1=NY+1
C
C   EVALUT
C   EVALUATION OF GRADIENT AT EACH POINT OF THE PICTURE
C
      DO 1 I=1,NX1
      DO 1 J=1,NY1
      K2=0
      IF(J.EQ.1) GO TO 30
      K1=0
      IF(PIC(I,J).EQ.PIC(I,J-1)) GO TO 31
      K1=PIC(I,J)-PIC(I,J-1)
   31 IF(K1.LT.CCT.AND.K1.GT.-CCT) GO TO 32
      IF(K1.LT.0) K2=2
      IF(K1.LT.0) K1=-K1
      KL=MADDON(I,J)
      CALL STORE(KL,K1,K2)
      MADDON(I,J)=KL
      IF(I.EQ.NX1) GO TO 32
       KL=MADDON(I+1,J)
      K2=K2+4*2
      CALL STORE(KL,K1,K2)
      MADDON(I+1,J)=KL
      GO TO 32
   30 K1=PIC(I,J)
      GO TO 31
   32 K2=1
      IF(I.EQ.1) GO TO 33
```

```
      K1=0
      IF(PIC(I,J).EQ.PIC(I-1,J)) GO TO 34
      K1=PIC(I,J)-PIC (I-1,J)
   34 IF(K1.LT.CCT.AND.K1.GT.-CCT) GO TO 1
      IF(K1.GE.0) K2=3
      IF(K1.LT.0) K1=-K1
      K2=4+K2
      KL=MADDON(I,J)
       CALL STORE(KL,K1,K2)
      MADDON(I,J)=KL
      IF(J.EQ.NY1)  GO TO 1
      KL=MADDON(I,J+1)
      K2=8+K2
       CALL STORE(KL,K1,K2)
       MADDON(I,J+1)=KL
      GO TO 1
   33 K1=PIC(I,J)
      GO TO 34
    1 CONTINUE
C
C       EDGE POINT . IDNS  KEEPS TRACK OF THE CONTOUR
C       DIRECTION AT ITS STARTING POINT
C   IDNI CONTAINS THE POSSIBLE DIRECTION OF THE NEXT
C   EDGE TRACING IS PERFORMED BY  THE FOLLOWING SECTION
C
      DO 20 I=1,NX1
      DO 20 J=1,NY1
      K=0
      INDX=0
      ISW=0
      IDNS=0
   18 CALL START(I,J,K,IDNS)
      IF(K.EQ.0.AND.ISW.EQ.1)GO TO 21
      IF(K.EQ.0) GO TO 20
       INDX=INDX+1
      L1=I
       L2=J
    3 X(INDX)=L1
       Y(INDX)=L2
      Z(INDX)=K
   25 K2=IDNI*4
      K1=0
      KL=MADDON(L1,L2)
      CALL STORE(KL,K1,K2)
      MADDON(L1,L2)=KL
      I1=L1
      I2=L2
      IDNI=IDNI+1
      GO TO (132,130,133,131),IDNI
  134 KL=MADDON(I1,I2)
      IDNI=IDNI-1
      K2=IDNI+2
      IF(K2.GE.4) K2=K2-4
      K2=K2*4
```

```
      K1=0
       CALL STORE(KL,K1,K2)
      MADDON(I1,I2)=KL
      PIC(L1,L2)=1.
      PIC(I1,I2)=1.
      IF(ISW.EQ.1) GO TO 6
    2 L1=I1
      L2=I2
      IJK=0
      CALL START(L1,L2,K,IJK)
   26 IF(K.EQ.0) GO TO 10
      INDX=INDX+1
      IF(ISW.EQ.1.AND.INDX.LT.LX) GO TO 25
      IF(INDX.LT.LX) GO TO 3
      X(INDX)=L1
      Y(INDX)=L2
      Z(INDX)=K
    8 FORMAT(I8)
    9 FORMAT(/6(2I5,I8,3X)/)
    7 FORMAT(24I3)
      PRINT 8,INDX
      PRINT 9,(X(L),Y(L),Z(L),L=1,INDX)
      PUNCH 8,INDX
      PUNCH 7,(X(L),Y(L),Z(L),L=1,INDX)
      INDX=1
      ISW=2
      GO TO 3
  130 I2=L2+1
      GO TO 134
  131 I2=L2-1
      GO TO 134
  132 I1=L1+1
      GO TO 134
  133 I1=L1-1
      IF(I1.EQ.0) PRINT 211,MADDON(L1,L2)
  211 FORMAT(I15)
      GO TO 134
   10 IF(INDX.GT.1) GO TO 4
      ISW=0
      IF(IDNS.LT.4) GO TO 18
      GO TO 20
    4 IF(ISW.EQ.0) GO TO 5
   21 ISW=0
      PRINT 8,INDX
      PRINT 9,(X(L),Y(L),Z(L),L=1,INDX)
      PUNCH 8,INDX
      PUNCH 7,(X(L),Y(L),Z(L),L=1,INDX)
      INDX=0
      IF(IDNS.LT.4) GO TO 18
      GO TO 20
    5 ISW=1
      L1=I
      L2=J
      K=-Z(1)
```

```
      GO TO 18
    6 DO 17 LK=2,INDX
      LK1=INDX-LK+2
      X(LK1)=X(LK1-1)
      Y(LK1)=Y(LK1-1)
   17 Z(LK1)=Z(LK1-1)
      X(1)=I1
      Y(1)=I2
      Z(1)=-K
      GO TO 2
   20 CONTINUE
      PRINT 11
   11 FORMAT(*0   OVER*)
      RETURN
      END
$IBFTC STORE
C
C  KEEPS TRACK OF THE NEIGHBOURING EDGE POINTS
C  THE RELEVENT INFORMATION IS CODED BY STORE
C
      SUBROUTINE STORE(KL,K1,K2)
      K=K2/4
      K3=K2-K*4
      K=K*8
      K3=K3*2**6+K1
      N=(2**K)*(2**8)
    1 KL=KL-KL/2**K*2**K+KL/N*N+K3*2**K
      RETURN
      END
$IBFTC START
C
C  CHECS FOR AN EDGE POINT ON THE CONTOUR
C  IND  - DIRECTION W.R.T PREVIOUS EDGE POINT
C  GRADIENT  OF THE NEW EDGE POINT
C
      SUBROUTINE START(I,J,K,IND)
      COMMON/CONT1/MADDON(50,65),X(100),Y(100)
      COMMON/PAR/IDNI,IDNS
      COMMON/CONT/NX,NY,CCT,CST
      INTEGER CCT,CST
    2 IND1=IND*8
      KL=MADDON(I,J)/2**IND1
      KL1=KL-KL/256*256
      KL2=KL1/2**6
       KL1=KL1-KL2*2**6
      IF(KL1.NE.0) GO TO 1
    5 IND=IND+1
       IF(IND.LT.4) GO TO 2
       GO TO 3
    1 IF(IND.NE.KL2) KL1=-KL1
      IF(KL1*K.LT.0) GO TO 5
      K=KL1
      GO TO 4
    3 K=0
```

```
    4 IDNI=IND
      RETURN
      END
C         **************************************************
C         *
C         *         CONTOUR  SMOOTHING
C         *
C         **************************************************
*IBFTC CONSMO
C
C  THIS ROUTINE DIVIDES THE CONTOURS INTO SEGMENTS WHEREIN
C      THE X-COORDINATE IS ALWAYS INCREASING.THAT IS , EACH
C      SEGMENT CAN BE REPRESENTED BY ASINGLE VALUED FUNCTION
C  X- INPUT ARRAY CONTAINING THE GIVEN CONTOUR
C  Y- OUTPUTARRAY CONTAINING THE SMOOTHED CONTOUR
C
      COMMON/PARA/L1,L2,L3
      COMMON/INO/X(3,100),Y(3,100)
      INTEGER X,Y
  807  READ 800,LX
C
C  L1,L2,L3, INDICATORS DEFINING THE STARTING AND ENDING
C      PORTIONS OF VARIOUS SEGMENTS
C
       READ 801,((X(I,J),I=1,3),J=1,LX)
  801 FORMAT(6(2I3,I6))
      L2=1
  800 FORMAT(I8)
      L1=0
      LX2=LX-1
  814 LX1=L1+1
      DO 812 I=LX1,LX2
      IF(X(1,I).GT.X(1,I+1)) GO TO 815
  812 IF(X(1,I).LT.X(1,I+1)) GO TO 825
      L1=LX
      CALL SHIFT(1)
  813 CONTINUE
      ILAST=1
      ISTART=1
      L=0
       K=100
      DO 802 I=1,L3
      IF(L.LT.1) GO TO 805
      IF(X(1,I).EQ.K) GO TO 802
      IF(ILAST.EQ.ISTART) GO TO
      IF(L.EQ.1) GO TO 805
      IMID=(ILAST+ISTART)/2
      DO 808 J=1,3
  808 Y(J,L)=X(J,IMID)
  805 L=L+1
      DO 803 J=1,3
  803 Y(J,L)=X(J,I)
      ISTART=I
  806 K=X(1,I)
```

```
  802 ILAST=I
      IF(ISTART.EQ.ILAST) GO TO 811
C     IMID=(ISTART+ILAST)/2
C     L=L+1
      IF(L.EQ.1) L=2
      DO 809 J=1,3
C     Y(J,L-1)=X(J,IMID)
  809 Y(J,L)=X(J,L3)
  811 CONTINUE
      PRINT 800,L
       PRINT 804,((Y(J,I),J=1,3),I=1,L)
  804 FORMAT(1H ,7(2X,2I4,I8))
       PUNCH 800,L
      PUNCH801,((Y(J,I),J=1,3),I=1,L)
      IF(L1.LT.LX-1) GO TO 814
      IF(L1.EQ.LX-1) GO TO 820
      GO TO 807
  815 N=2
      DO 816 I=LX1,LX2
  816 IF(X(1,I).LT.X(1,I+1)) GO TO 817
  819 L1=LX
      CALL SHIFT(N)
      GO TO 813
  817 L1=I
      CALL SHIFT(N)
      GO TO 813
  825 N=1
      DO 818 I=LX1,LX2
  818 IF(X(1,I).GT.X(1,I+1)) GO TO 817
      GO TO 819
  820 L=1
      DO 310 J=1,3
  810 Y(J,1)=X(J,LX)
      PRINT 800,L
      PRINT 804,((Y(J,I),J=1,3),I=1,L)
      PUNCH 800,L
      PUNCH 801,((Y(J,I),J=1,3),I=1,L)
      GO TO 807
       END
$IBFTC SHIFT
      COMMON/INO/X(3,100),Y(3,100)
C  THIS ROUTINE SHIFTS EACH SEGMENT TO PRE ASSIGNED LOCATIONS
      COMMON/PARA/L1,L2,L3
      INTEGER X,Y
      IF(L2.EQ.1) GO TO 2
      DO 1 I=L2,L1
      I1=I-L2+1
       DO 1 J=1,3
    1 X(J,I1)=X(J,I)
    2 L3=L1-L2+1
      L2=L1+1
```

```
      IF(N.EQ.1) RETURN
      L4=L3/2
      DO 4 J=1,3
      DO 4 I=1,L4
      N=L3-I+1
      Z=X(J,I)
      X(J,I)=X(J,N)
    4 X(J,N)=Z
      RETURN
      END
C          ****************************************
C          *                                      *
C          * SPLINE APPROXIMATION PROBLEM         *
C          *                                      *
C          ****************************************
$IBFTC APPROX
C
C     THIS DETERMINES THE MINIMUM NUMBER OF KNOTS
C     NEEDED FOR A GIVEN FIT AFTER OPTIMISING THEIR
C     POSITION ONLY THOSE ROUTINES WHICH HAVE BEEN
C     MODIFIED HAVE BEEN LISTED . FOR OTHERS REFER
C     TO REF.3 AND 4
C     XIL - POSITION OF KNOTS
C     ERRL2 - L2- ERROR
C     ERRL99 - MINMAX ERROR
C     FOR OTHER DETAILS REFER TO REF. 3 AND 4
C
      COMMON/INPUT/LX,XX(50),U(50),JADD,ADDXI(20),MODE
      COMMON/OUTPUT/UERROR(50),FCTL(50),XIL(22),COEFL(21,4),
     *VORDL(22,2),KNOT,LMAX,INTE9V
      COMMON /OTHER/LXI,LXI1,LXI2,Q,CHANGE,ERROR,ACC,XI(28)
      COMMON/CONS/PHI2
      COMMON/OUT2/ERRL1,ERRL2,ERRL99
      COMMON/INTERN/Y1(50),Y2(50),Y3(50)
      COMMON/ADDN/KX
      DATA BLANK/2H  /
      INTEGER Y1,Y2,Y3
      PHI2=2.*ATAN(1.)
      GIVER=1.5
      GO TO 1705
    1  CONTINUE
 1705 ACC=0.1
 1702 FORMAT(I8,A2)
      READ 1702,LX,AMARK
      KX=0
      KNTMAX=FLOAT(LX)/3.5
      IF(LX.LE.7) KNTMAX=LX-2
      IF(LX.GT.7.AND.LX.LE.12) KNTMAX=LX/2
 1701 FORMAT(24I3)
      READ 1701,(Y1(I),Y2(I),Y3(I),I=1,LX)
      IF(KNTMAX.GE.3) GO TO 23
      PRINT 1706,LX
 1706 FORMAT(//* DATA NOT SUITABLE FOR APPROXIMATION */* NO. OF POIN
     1 *,I5)
```

```
      PUNCH 2702,LX
 2702 FORMAT(I8)
      PUNCH 1701,(Y1(I),Y2(I),Y3(I),I=1,LX)
      GO TO 1705
   23 DO 1703 I=1,LX
      XX(I)=Y1(I)
 1703 U(I)=Y2(I)
      WRITE(6,611) (I,XX(I),U(I),I=1,LX)
      ADDXI(1)=XX(1)
      ADDXI(2)=XX(LX)
      XI(1)=XX(1)
      XI(2)=XX(LX)
      IF(AMARK.EQ.BLANK)     GO TO 12
   15 FORMAT( I8,/12(I3,3X))
      READ(5,15) LXI,(Y2(I),I=1,LXI)
      IKIN=LXI+3
      MODE=0
      JADD=IKIN-1
      DO 16 I=3,JADD
      XI(I-1)=Y2(I-2)
   16 ADDXI(I-1)=XI(I-1)
      XI(JADD)=XX(LX)
      ERROR=FXDKNT(2.)
      GO TO 18
   12 IKIN=3
      MODE=0
      JADD=2
      ERROR=FXDKNT(2.)
   18 DO 9 I=IKIN,KNTMAX
      ITER=3
      LXI2=I
      ADDXI(1)=XX(LMAX)
      LXI1=LXI2-1
      LXI=LXI1-1
      DO 2J=1,LXI1
    2 IF(XX(LMAX).LT.XI(J)) GO TO 3
      J=LXI1
      IF(XX(LMAX).EQ.XI(LXI1)) GO TO 3
      WRITE(6,700) LMAX,LXI1
  700 FORMAT(* LMAX OUT OF BOUNDS. ITS VALUE = *,I3,*NO. OF KNOTS
     1UCED *,I3)
   51 IF(ERRL99.GT.(2.5*GIVER)) GO TO 27
      IKAX=FLOAT(KNTMAX)/1.5
      IF(LXI1.GT.IKAX) GO TO 7
      GO TO 27
C   3 IF(ABS(XI(J)-XX(LMAX)).LT.1.0) GO TO 20
C     IF(ABS(XX(LMAX)-XI(J-1)).GE.1.0) GO TO 5
C  20 IF(ABS(XI(J)-XI(J-1)).LT.2.0) GO TO 6
    3 ADDXI(1)=INT((XI(J)+XI(J-1))/2.0)
      IF(ABS(XI(J)-XI(J-1)).LT.2.) GO TO 6
    5 DO 4 K=J,LXI1
      K1=LXI2-K+J
    4 XI(K1)=XI(K1-1)
      XI(J)=ADDXI(1)
```

```
      KNOT=J
  850 FORMAT(* LMAX,XX(LMAX),ADDXI(1) ARE PRINTED BELOW*/22X,I6,2F12
      DO 8 L=J,LXI1
      L1=L-J+1
    8 ADDXI(L1)=XI(L)
      MODE=1
      JADD=LXI2-J
   11 ERROR=FXDKNT(0.)
      CALL SWEEP(ITER)
      MODE=1
      JADD=0
      DUMB=FXDKNT(2.)
      IF(ERRL99.GT.GIVER) GO TO 9
      DUMB=FXDKNT(-1.)
      IF(ERRL99.LE.GIVER) GO TO 7
      GO TO 9
    6 WRITE(6,620) I,XI(J),XI(J-1)
  620 FORMAT(* OPTIMIZATION DISCONTINUED DUE TO CROWDING OF KNOTS*//
     1ILE INTRODUCING *,I3,* -TH KNOT*//* VALUES OF ADJACENT
     2KNOTS = *,2F12.8)
      GO TO 51
   27 XI(LXI2)=XX(LX)
      LXI1=LXI2-1
      LXI=LXI1-1
      DEL=(XX(LX)-XX(1))/FLOAT(LXI1)
      DO 26 J=3,LXI2
      XI(J-1)=INT(XI(J-2)+DEL)
   26 ADDXI(J-2)=XI(J-1)
      MODE=1
      KNOT=2
      JADD=LXI
       DO 28 J=1,LXI
   28 ADDXI(J)=XI(J+1)
      DUMB=FXDKNT(0.)
      CALL SWEEP(ITER)
      MODE=1
      JADD=0
      DUMB=FXDKNT(2.)
    9 CONTINUE
      WRITE(6,750) KNTMAX
    7 MODE=1
   10 WRITE(6,640)
      JADD=0
      DUMB=FXDKNT(1.)
      GO TO 1
  610 FORMAT(I8/(24I3))
  611 FORMAT(* GIVEN DATA*//5(I4,2F10.5))
  640  FORMAT(49X,* FINAL OUTPUT*///)
  750 FORMAT(*1 GIVEN ACCURACY CAN NOT BE OBTAINED EVEN WITH *,I3,*
     1S    *)
      END
C
C     THIS ROUTINE SOLVES THE FIXED KNOT PROBLEM
C     Y1,Y2    CONTAIN THE APPROXIMATED CONTOURS
```

```
C     FOR DETAILS ON OTHER VARIABLE SEE REF) 3
C     THE OPTION (-1) FOR ARG QUANTIZES THE END
C     DERIVATES . VORDL(2,2) , VORDL(KNOT,2)
C     CONTAINS THE END DERIVATIVES
C
$IBFTC FXDKNT
      FUNCTION FXDKNT(ARG)
      DOUBLE PRECISION TRPZWT,SUM
      LOGICAL MODE3
      DIMENSION WEIGHT(50),CUBERR(50)
      COMMON /WANDT/TREND(50) ,TRPZWT(50) ,PRINT(100)
      COMMON /INPUT/LX,XX(50) ,U(50) ,JADD,ADDXI(20),MODE
      COMMON/OUTPUT/UERROR(50) ,FCTL(50) ,XIL(22),COEFL(21,4),
     *VORDL(22,2),KNOT,LMAX,INTERV
      COMMON /BASIS/FCT(50,25) ,VORD(30,23,2),BC(25),ILAST
      COMMON/LASTB/IORDER(28),INSIRT(30),XKNOT
      COMMON/OUT2/ERRL1,ERRL2,ERRL99
      COMMON/INTERN/Y1(50),Y2(50),Y3(50)
      INTEGER Y1,Y2,Y3
      EQUIVALENCE (IPRINT,CHANGE)
      CHANGE=ARG
      IF(MODE.GT.0)  GO TO 29
      XSCALE=XX(LX)-XX(1)
      DO 10 I=5,30
   10 INSIRT(I)=0
      DO 11 L=1,LX
      UERROR(L)=U(L)
      TREND(L)=T(XX(L))
   11 WEIGHT(L)=W(XX(L))
      DO 12 L=2,LX
   12 TRPZWT(L)=(XX(L)-XX(L-1))/4.*(WEIGHT(L-1)+WEIGHT(L))
      XIL(1)=ADDXI(1)
      XIL(2)=ADDXI(2)
      IORDER(1)=1
      IORDER(2)=2
      KNOT=2
      INTERV=1
      DO 19 I=1,4
      ILAST=I
      CALL NUBAS
      DO 19 L=1,LX
   19 UERROR(L) =UERROR(L)-BC(I)*FCT(L,I)
      MODE=1
      DO 20 L=1,LX
   20 CUBERR(L)=UERROR(L)
      JADD=JADD-2
      IF(JADD.LE.0) GO TO 60
      DO 21 I=1,JADD
   21 ADDXI(I)=ADDXI(I+2)
      GO TO 51
   29 GO TO (40,40,30),MODE
   30 XKNOT=CHANGE
      IF(MODE3) GO TO 35
      MODE3=.TRUE.
```

```
      ERBUT1=FXDKNT
      MODE=2
      CALL NUBAS
      KNOTSV=KNOT
      MODE=3
      GO TO 36
   35 CALL NUBAS
   36 FXDKNT=ERBUT1-BC(ILAST)/XSCALE*BC(ILAST)
      RETURN
   40 IF(KNOT.LT.KNOTSV) GO TO 42
      KNOT=KNOTSV
      IF(.NOT.MODE3) GO TO 50
      DO 41 L=1,LX
   41 UERROR(L)=UERROR(L)-BC(ILAST)*FCT(L,ILAST)
      GO TO 49
   42 DO 43 L=1,LX
   43 UERROR(L)=CUBERR(L)
      IF(KNOT.LE.2) GO TO 48
      IDUM=KNOT+1
      DO 45 IO=IDUM,KNOTSV
      INSERT=INSIRT(ILAST)
      ILM3=ILAST-3
      DO 44 K=INSERT,ILM3
      IORDER(K)=IORDER(K+1)
   44 XIL(K)=XIL(K+1)
   45 ILAST=ILAST-1
      DO 47I=5,ILAST
      DO 47 L=1,LX
   47 UERROR(L)=UERROR(L)-BC(I)*FCT(L,I)
       GO TO 49
   48 XIL(2)=XIL(ILAST-2)
      IORDER(2)=2
       KNOT=2
   49 IF(JADD.GT.0) GO TO 51
      ILAST=KNOT+2
      INTERV=KNOT-1
      GO TO 60
   50 IF(JADD.LE.0) GO TO 61
   51 DO 52 IO=1,JADD
      XKNOT=ADDXI(IO)
      CALL NUBAS
      DO 52 L=1,LX
   52 UERROR(L)=UERROR(L)-BC(ILAST)*FCT(L,ILAST)
   60 FXDKNT=DOT(31,2)/XSCALE
      KNOTSV=KNOT
   61 MODE3=.FALSE.
      IF(IPRINT.LT.0) IPRINT=-IPRINT
      IF(IPRINT.EQ.0) RETURN
      GO TO (70,80,90),IPRINT
   70 WRITE(6,610)
      DO 72 I=1,KNOT
      ILOC=IORDER(I)
      DO 72 L=1,2
      SUM=0.D0
```

```
      DO 71 J=1,ILAST
   71 SUM=SUM+BC(J)*VORD(J,ILOC,L)
   72 VOPDL(I,L)=SUM
      DO 75 IN=1,KNOT
   75 VORDL(IN,1)=INT(VORDL(IN,1))
      A1=VORDL(1,2)
      A2=VOPDL(KNOT,2)
      LL1=2
      NQU=128/LL1-1
      CALL QUANT(VORDL(1,2),VORDL(LX,2))
      CALL INTERP
      CALL EVAL
      DO 76 IN=1,LX
   76 UERROR(IN)=FLOAT(Y2(IN))-U(IN)
      IF(ARG.LT.0) GO TO 85
      FXDKNT=DOT(31,2)/XSCALE
  801 FORMAT(// 1H ,15X*NO. OF QUANTISATION LEVELS
     1  = *,F12.6,15XF12.6)
      DO 73 I=1,KNOT
   73 PRINT 620,I,XIL(I),VORDL(I,1)
  610 FORMAT(42X,5HKNOTS,22X,* VALUES   *//)
  620 FORMAT(35X,3HXI(,I2,3H)= ,F12.6,22X,F12.6)
      PRINT 630,VORDL(1,2),VORDL(KNOT,2)
  630 FORMAT(*0*,35X,* END DERIVATIVES*,* = *,F12.6,15X,F12.6)
   80 ERPL2 = SQRT(FXDKNT)
      ERRL1 = 0.
      ERRL99=0.
      DO 82 L=1,LX
      DIF =ABS(UERROR(L)*WEIGHT(L))
      IF(ERRL99.GT.DIF) GO TO 81
      LMAX=L
      ERRL99=DIF
   81 ERRL1=ERRL1+DIF
      IF(ARG.LT.0) RETURN
   82 CONTINUE
      ERRL1=ERRL1/FLOAT(LX)
      GO TO (90,96,96),IPRINT
   90 CONTINUE
      PRINT 100,LMAX,ERRL99 ,ERRL1,ERRL2
  100 FORMAT(1H ,*LMAX,ERRL99 = *,53X,I5,F12.6/
     1*0ERRL1,ERRL2 = *,45X2(F12.6,10X))
      PRINT 800,(XX(L),FCTL(L),L=1,LX)
  800 FORMAT(*0FINAL VALUES OF XX,FCTL */(2X,8F16.8))
      VORDL(KNOT(1,2)=A1
      VORDL(KNOT,2)=A2
   96 RETURN
      END
C
C     THIS ROUTINE QUANTIZES THE ANGLE OF
C     INCLINATION TO THE REQUIRED FINENESS
C     N - NUMBER OF LEVELS
C     A,B - QUANTITIES TO BE QUANTISED
C
$IBFTC QUANT
```

```
      SUBROUTINE QUANT(A,B,N)
      COMMON/CONS/PHI2
      AN=N
      AN=AN/PHI2
      AN=AN/2.
      A1=ATAN(A)
      IF(ABS(A1).GT.(PHI2*2./3.)) AN=4.*AN
      K1=ABS(A1*AN)+0.5
      A2=FLOAT(K1)/AN
       IF(ABS(A2-PHI2).LT.0.002) A2=PHI2-0.002
      IF(A1.LT.0.) A2=-A2
      A=TAN(A2)
      AN=N
      AN=AN/PHI2
      AN=AN/2.
      B1=ATAN(B)
      IF(ABS(B1).GT.(PHI2*2./3.)) AN=4.*AN
      K2=ABS(B1*AN)+0.5
      B2=FLOAT(K2)/AN
       IF(ABS(B2-PHI2).LT.0.002) B2=PHI2-0.002
      IF(B 1.LT.0.) B2=-B2
      B=TAN(B2)
C            ****************************************
C            *                                      *
C            *    PICTURE RECONSTRUCTION            *
C            *                                      *
C            ****************************************
$IBFTC *MAN*
      COMMON/MAIN/PIC(65,65)
      COMMON/FILT/PICH(65,65),FIL1(25),FIL2(25),HX(12,12)
      COMMON/PAR/NX,NY,N
      COMMON/OUTS/ALIN(65),IGR(2)
      INTEGER PIC,PICH,FIL1,FIL2,HX
      DIMENSION FILL(25)
      INTEGER X(100),Y(100),Z(100)
      INTEGER PICINT(65,65)
      NX=41
      NY=45
      N=10
      FELE=0
       SUM=0.
      SIGMA=4.0
      NM=NX
      IF(NM.LT.NY) NM=NY
      DO 152 I=1,NM
      DO 152 J=1,NY
      PICH(I,J)=0
       PICINT(I,J)=0
  152 PIC(I,J)=0
C
C     H P F RESPONSE GENERATION
C
      DO 4 I=1,N
      R=(I-1)*(I-1)
      R=R/(2.*SIGMA)
```

```
      FILL(I)=1./EXP(R)
    4 SUM=SUM+FILL(I)
      SUM=2.*SUM-1.
      DO 5 I=1,N
      FIL1(I)=FILL(I)*1.E03/SUM
    5 FIL2(I)=FIL1(I)
      DO 505 I=1,N
  505 HX(N,I)=0
      DO 501 J=2,N
      I=N-J+2
      IF(I.EQ.N) GO TO 504
      I1=I+1
      DO 502 L1=I1,N
  502 HX(I-1,L1)=0
  504 HX(I-1,I)=(FIL1(I)*FIL2(I)+2*HX(9,I))/2
      HX(I-1,I-1)=FIL1(I)*FIL2(I-1)+HX(I,I-1)+HX(I-1,I)
      IF(I.EQ.2) GO TO 503
      I2=I-2
      DO 501 L=1,I2
      K=I-1-L
  501 HX(I-1,K)=FIL1(I)*FIL2(K)+HX(I,KC
  503 HX(1,1)=(10 0000-FIL1(1)*FIL2(1)8/4
   75 FORMAT(I8)
   80 READ 75,LX
      IF(LX.EQ.0) GO TO 78
      READ 76,(X(I),Y(I),Z(I),I=1,LX)
   76 FORMAT(6(2I3,I6))
      DO 77 I=2,LX
      K1=X(I)
      K2=Y(I)
      K=X(I-1)
       L=Y(I-1)
      K1=X(I-1)-X(I)
      K2=Y(I-1)-Y(I)
      KL=Z(I-1)
      IF(IABS(K1).GT.1.OR.IABS(K2).GT.1) GO TO 23
      IF(IABS(K1*K2).EQ.1) GO TO 23
      IF(K1.EQ.0) GO TO 21
      IF(51.EQ.1.AND.K2.EQ.0) L=L-1
      IF(K1.EQ.(-1).AND.K2.EQ)0) KL=-KL
      PIC(K,L)=KL
      GO TO 77
   21 IF(K2.EQ.(-1)) GO TO 22
      K=K-1
      KL=-KL
   22 PICINT(K,L)=KL
   77 CONTINUE
      GO TO 80
   78 CONTINUE
      IND1=1
   85 CONTINUE
      CALL HIFIL
      DO 92 I=1,NM
```

```
      DO 92 J=1,NM
   92 PIC(I,J)=0
      IF(IND1.EQ.2) GO TO 90
      DO 83 I=1,NM
      IF(I.EQ.NM) GOTO 84
      J1=I+1
      DO 83 J=J1,NM
      K=PICH(I,J)
      PICH(I,J)=PICH(J,I)
      PICH(J,I)=K
      K=PICINT(I,J)
       PICINT(I,J)=PICINT(J,I)
   83 PICINT(J,I)=K
   84 IND1=IND1+1
      DO 25 I=1,NM
      DO 25 J=1,NM
      PIC(I,J)=0
   25 PIC(I,J)=PICINT(J,I)
        NY=NX
      NX=NM
      GO TO 85
   90 NX=NY
      NY=NM
      READ 79,((PIC(I,J),J=1,NY),I=1,NX)
C  79 FORMAT(40I2)                     TAGORE
   79 FORMAT(40I2)
      DO 86 I=1,NM
      DO 86 J=1,NY
      PICH(J,I)=PICH(J,I)/1000000
   86 PIC(I,J)=PIC(I,J)+PICH(J,I)
      DO 251 I=1,6
      PRINT 161
  161 FORMAT(1H1//////)
      CALL OUTPUT(1)
      PRINT 162
  162 FORMAT(////30X,* FIG 4.11    RECEIVED PICTURE - GRAHAMS
  251 CONTINUE
      PUNCH 79,((PIC(I,J),J=1,NY),I=1,NX)
      STOP
      END
C          ****************************************
C          *    HIGH PASS FILTER                  *
C          *                                      *
C          ****************************************
$IBFTC HIFIL
      SUBROUTINE HIFIL
      COMMON/MAIN/PIC(65,65)
      COMMON/FILT/PICH1(65,65),FIL1(25),FIL2(25),HX(12,12)
      COMMON/PAR/NX,NY,N
      COMMON/FILT2/FELE,FCOL(65),FROW(65)
      INTEGER GRADX,PICH1,FIL1,FIL2,HX
      INTEGER FELE,FCOL,FROW
```

```
      DO 30 I=1,NX
      DO 30 J=1,NY
      N1=I
      IF(I.GT.N) N1=N
      N2=J
      IF(J.GT.N) N2=N
      DO 43 K=1,N1
      I1=I-K
      DO 35 L=1,N2
      J1=J-L
   35 PICH1(I,J)=PICH1(I,J)+HX(K,L)*PIC(I1+1,J1+1)
   34 N3=NY-J
      IF(N3.GE.N) N3=N-1
      IF(N3.LE.0) GO TO 43
      DO 37 L=1,N3
      L1=L+1
      L2=J+L-1
   37 PICH1(I,J)=PICH1(I,J)+HX(K,L1)*PIC(I1+1,L2+1)
   43 CONTINUE
      N4=NX-I+1
      IF(N4.GT.N) N4=N
      IF(N4.LE.0) GO TO 30
      DO 38 K=1,N4
      K1=K
      K2=I+K-1
      DO 39 L=1,N2
      J1=J-L
   39 PICH1(I,J)=PICH1(I,J)-HX(K1,L)*PIC(K2+1,J1+1)
   42 IF(N3.LE.0) GO TO 38
      DO 40 L=1,N3
      L1=L+1
      L2=L+J-1
   40 PICH1(I,J)=PICH1(I,J)-HX(K1,L1)*PIC(K2+1,L2+1)
   38 CONTINUE
   30 CONTINUE
      RETURN
      END
   23 PRINT 4,K1,K2,K
    4 FORMAT(* POINTS NOT CONSECUTIVE*/* K1,K2,K = *,3I6)
      STOP

C              END OF SIMULATION

C              OVER
```